Multi-domain routing in Delay Tolerant Networks

Alan Hylton
NASA Goddard
alan.g.hylton@nasa.gov

Brendan Mallery
Tufts University
Brendan.mallery@tufts.edu

Jihun Hwang
Purdue University
hwang102@purdue.edu

Mark Ronnenberg
Indiana University
maronnen@iu.edu

Miguel Lopez
University of Pennsylvania
mlopez3@sas.upenn.edu

Oliver Chiriac
University of Oxford
oliver.chiriac@trinity.ox.ac.uk

Sriram Gopalakrishnan
Sorbonne Université
sriram.gopalakrishnan@lip6.fr

Tatum Rask
Colorado State University
tatum.rask@colostate.edu

Abstract—**The goal of Delay Tolerant Networking (DTN) is to provide the *missing ingredient* for the ever-growing collection of communicating nodes in our solar system to become a Solar System Internet (SSI). Great strides have been made in modeling particular types of DTNs, such as schedule- or discovery-based. Now, analogously to the Internet, these smaller DTNs can be considered routing domains which must be stitched together to form the overall SSI. In this paper, we propose a framework for cross-domain routing in DTNs as well as methodologies for detecting these sub-domains. Example time-varying networks are given to demonstrate the techniques proposed.**

A basic component is the mathematical theory of sheaves, which unifies the underlying model of DTN routing algorithms, by giving rise to *routing sheaves* – these can be defined for the dynamic and scheduled networks as noted above, and can also be used to define the interfaces between these domains in order to route across them. An immediate application would be routing across discovery-based networks connected by scheduled networks.

These DTN subdomains remain elusive, however, and need to become well-defined and properly sized for tractable computability. In particular, a balance must be determined between areas that are too large (i.e. large matrix computations) versus areas that are too small (i.e. "many" single-noded domains). Moreover, the connections between the domains should, at least locally, be chosen to optimize data flow and connectivity: we address this in three ways. First, tools from persistent homology are given to understand underlying structures, reminiscent of hierarchies in the Internet Protocol (IP) addressing. Second, we construct a notion of temporal graph curvature based on network geometry to analyze flows induced by dynamical processes on these networks. Finally, Schrödinger Bridges, a tool arising from statistical physics, are proposed as a method of constructing flows on time-evolving networks with desirable properties such as speed, robustness, and load sensitivity.

We construct an approach to temporal hypergraphs to simultaneously model unicast, multicast, and broadcast, using the language of scheme theory, and then consider DTN network coding as a way to achieve network-level computation and organization. The paper concludes with a discussion and ideas for future work.

TABLE OF CONTENTS

1. INTRODUCTION

The Internet is an archetypal scalable system. This status is due to an ever-growing base of billions of users. The Internet's ability to achieve such a positive returns to scale comes from decades of study and development into its structures, for example autonomous systems, addressing, domains, subnetworks, etc. Having these well-defined structures allows, for example, policy-based inter-domain routing like the Border Gateway Protocol (BGP), which itself is a scalability-enabler. With accelerating growth of space systems as well as the complexity of these systems, it becomes necessary to generalize Internet-like capability to the embryonic Solar System Internet (SSI). Consider NASA's LunaNet [1], which features requirements for multi-hop multi-path space networks.

The key difficulty is that the SSI is a vast generalization of terrestrial networks. These reasons are well-documented, but we note that:

- link latencies may vary between milliseconds and hours,
- links are almost never symmetric,
- end-to-end paths might never exist,
- many communication opportunities are scheduled, and
- there is no addressing scheme.

The comment element here is that the temporal nature of the network is paramount. As such, other missing ingredients include definitions of *temporal* domains, *temporal* autonomous systems, and so forth.

Returning to structure, we emphasize that the graph-of-graphs approach to the Internet has been essential for its own scalability, that is, the ability to model the Internet at various levels is vital. For the SSI, we must also have means to model analogous constituents of the space network and be able to route between them.

The goal of this paper is to explore the missing definitions noted, adding maturity to them, and then discussing how to assemble these pieces into a network. It must be emphasized that in a true SSI, it is not possible to have global omniscience. For example, a network about the Earth (say in Low Earth Orbit) could discover most of its own network state, but any information it had about a network about Mars would

necessarily be minutes out of data at best. This shows why the ability to take routing domains and find and exploit their interfaces to each other is necessary for true scalability.

We cover three vastly different ways of modeling and detecting temporal network structure. A zeroth method is graph sequences, where at each time step there might be a change in vertices, edges, or both. Going a step further, we generalize graph curvature to the temporal setting. Using this tool, it can be shown that there are graph sequences where each graph in the sequence can have positive curvature yet the overall temporal curvature is negative. This means that the temporal network is greater than the sum of its parts, illustrating the need for better modeling. Persistent homology, a computational method that yields qualitative information about the network's connectivity, clusters, and other topological information, is also explored. Like curvature, persistence can also help detect well- and poorly- connected assets, communities, and otherwise help diagnose a network. Finally, we discuss how representing graphs using networks embeddings can help facilitate link prediction, communities, and even help classify the roles (e.g. router, data source, data sink).

Once a network has been modeled, the mathematical language of *sheaves* can be used to model information over the model; this includes routing [2]. Sheaves are data structure that are designed to track local data and either extend these local observations to global ones or show why this cannot be done. In this paper, we explore a new type of sheaf that carries additional structure to existing path-finding sheaves, the so-called lattice sheaves, and show how the additional data allow more germane interpretations in the context of communications networks. Because sheaves force consistency on interfaces between locales, there is also a theory of "gluing" different sheaves together into a new sheaf, even if these sheaves sit above different networks. This means that sheaves can make rigorous the notion of a gateway router - a necessary step for multidomain routing.

Finally, we consider multiple avenues for future work based on our results. These include some refinements for the multidomain routing sheaves, a method to bring the powerful machinery of algebraic geometry into play, and a discussion on network coding in the context of the SSI.

2. Curvature of Temporal Networks

Motivation

A temporal metric space is a metric space (such as a network, $\mathbb{R}^d$, a manifold) that evolves over time. The goal of this section is to establish a form of geometry on temporal metric spaces that captures the dynamics of communication, i.e. a functional on the temporal metric space that measures how long it takes to pass a message from one point to another. Even defining such a metric is not straightforward, as one needs to account for metric and even topological changes. We will describe one possible construction, and show that the resulting theory is rich enough to define a notion of curvature to this setting. We will then sketch out applications of this curvature to analyzing and designing time-evolving networks. We will keep the formal proofs of claims to a minimum, and will prioritize intuition. The full proofs will be given in a forthcoming work.

Temporal metric spaces and the geometry of communication:

Definition 2.1. Let X be a set. Then an (extended) metric is a function $\rho : X \times X \to \mathbb{R}_{\geq 0} \cup \{+\infty\}$ satisfying:

1. $\rho(x, x) = 0$ for all $x \in X$;
2. $\rho(x, y) = \rho(y, x)$ for all $x, y \in X$;
3. $\rho(x, y) \leq \rho(x, z) + \rho(z, y)$ for all $x, y, z \in X$.

We call a pair (X, ρ) a metric space.

Definition 2.2. Let X be a set. Then a temporal metric space is a function $X^t : \mathbb{R} \to \mathcal{MS}(X)$, where $\mathcal{MS}(X)$ denotes the set of metric spaces over X.

This definition subsumes the usual definition of metric space:

Definition 2.3. Let X^t be a temporal metric space such that X^t is constant as a function of t. Then we call X^t a static metric space.

Our primary example of a temporal metric space is given by a temporal network.

Definition 2.4. A temporal network is a function $\Gamma^t : \mathbb{R} \to \mathcal{G}_n$ from $\mathbb{R}$ to the set of possible graphs $\mathcal{G}_n$ on a labelled vertex set V.

Any graph Γ is equipped with an extended metric ρ, where $\rho(x, y)$ is the length of the shortest path from x to y in Γ. As our primary interest is modeling satellite networks, for the remainder of this section we will only deal with time-evolving networks. However, we remark that the results in this section also apply to the more general notion of temporal metric spaces. We also make the simplifying assumption that all graphs are unweighted.

To define a geometry to describe communication, we must first define a reasonable model of message passing. In a static network, one could achieve this by fixing some unit of time $r > 0$ and claiming that it takes r time to pass a message along an edge. In this way, the amount of time it takes to communicate between two nodes in the network is a constant multiple of the distance between the two nodes, and hence in this (albeit simple) model, the study of message-passing reduces to the geometry of the network. We will now define a model of communication over a time-evolving network that will recover the above model in the case of a static network.

Definition 2.5. Let Γ^t be a temporal network, let $x \in V$ and let $t_0 \in \mathbb{R}$. For any $K \in \mathbb{R}_{\geq 0}$, denote by $B^t_K(x)$ to be the set of nodes which are within distance K of x at time t_0 in Γ^{t_0}, i.e.:

$$B^{t_0}_K(x) = \{y \in \Gamma^{t_0} | \rho^{t_0}(x, y) \leq K\}. \tag{1}$$

Definition 2.6. Let Γ^t be a temporal network, let $x \in V$ and let $t_0 \in \mathbb{R}$. Then we define the following function from $\mathbb{R}$ to the set of subsets of V :

- For $s \in [t_0, t_0 + r)$, let $C^s_r(x, t) = \{x\}$;
- For all $\ell \in \mathbb{N}$, let $t + \ell r \leq s < t + (\ell + 1)r$, recursively define

$$C^s_r(x, t_0) = \cup_{y \in C^{t_0+\ell r}_r(x,t_0)} B^{t_0+\ell r}_1(x).$$

We refer to $C_r^s(x,t_0)$ as the r-cone at (x,t_0) at time s.

The definition of r-cones is designed to suggest the following model of communication: Suppose any node x at a time t_0 can communicate with any neighboring node (i.e. nodes that are distance 1 from x at time t_0) but that it takes r time for the message to reach a node once it is sent. Once a node y receives the message, the node can pass this message along to any neighbor in its own neighborhood at time t_0+r, and it once again takes r time for the message to be received. Then $C_r^s(x,t_0)$ defines the set of nodes that have received this message, originating from x at time t_0, at time s.

With these definitions established, we may now define our temporal metric:

Definition 2.7. Let Γ^t be a temporal network, and let $r>0$. Then for any $x,y\in V$ and $t_0\in\mathbb{R}$, we define:

$$d_r^{t_0}(x,y)=\inf_{s\in\mathbb{R}}\{s|y\in C_r^s(x,t_0)\}. \tag{2}$$

We call d_r^t the r-temporal cost functional on Γ^t.

In the aforementioned model for communication, the temporal cost represents the amount of time needed for x at time t_0 to pass a message to y. Much more can be said about this functional, but we limit ourselves to the following basic facts:

Proposition 2.1. *Let Γ^t be a temporal network and $r>0$. Then:*

1. $d_r^t(x,y)=0$ iff $x=y$.

2. d_r^t is not symmetric nor does it satisfy the triangle inequality.

3. d_t^r is lower semicontinuous.

4. If Γ^t is a static temporal network, then $d_1^t(x,y)=\rho(x,y)$ on Γ^{t_0} for any $t_0\in\mathbb{R}$.

The final statement in Proposition 2.1 shows that d_r^t simultaneously generalizes the model for communication laid out for static networks and the geometry of these networks. We now use this cost functional to define a notion of curvature for temporal networks.

Optimal transportation and curvature:

In [3], a notion of Ricci curvature for metric measure spaces was defined using the theory of optimal transport. One feature of this is that it allows one to define a geometric invariant of graph, which has generated a lot of interest from the network science, graph theory and combinatorics communities [4–7]. In this section we give a minimal introduction to the subject for graphs, pointing the reader to Ollivier's original paper for a more thorough introduction.

Definition 2.8. Let Γ be a graph, and let $\mathcal{P}(V)$ be the set of probability measures over the vertex set V. Let $P:\Gamma\to\mathcal{P}(V)$ be the map sending each $x\in\Gamma$ to the uniform probability measure on the neighborhood of x. We call P the uniform kernel on Γ.

Observe that for a finite graph, P may be represented as a matrix with rows and columns indexed by the vertices of Γ. Also notice that P generates a (lazy) simple random walk on Γ. Ollivier-Ricci curvature is a function that depends on the optimal transportation distance between probability measures defined by P on Γ, which we now introduce.

Definition 2.9. Let μ,ν be two probability measures on a graph Γ. Then we write $\Pi(\mu,\nu)$ to mean the set of couplings between μ and ν, i.e. the set of probability measures ζ on $\Gamma\times\Gamma$ such that the first marginal of ζ is μ and the second marginal is ν.

Example 2.1. Let $\mu,\nu\in\mathcal{P}(V)$. Then the product measure $\mu\times\nu$ is a coupling in $\Pi(\mu,\nu)$.

Definition 2.10. Let Γ be a graph, let $\mu,\nu\in\mathcal{P}(V)$ and let d denote the metric on Γ. Then the 1-optimal transport distance between μ and ν is:

$$W(\mu,\nu)=\inf_{\zeta\in\Pi(\mu,\nu)}\int d(x,y)d\zeta(x,y). \tag{3}$$

Intuitively, for a given $\zeta\in\Pi(\mu,\nu)$, $\int d(x,y)d\zeta(x,y)$ represents the average cost of moving mass distributed according to μ to mass distributed according to ν on the graph Γ according to the distribution ζ, where cost is determined by the distance. Optimal transport is then the minimal cost over all possible "transport plans" $\Pi(\mu,\nu)$.

Definition 2.11. Let Γ be a graph, and let P denote the uniform kernel on Γ. Then for any $x,y\in\Gamma$, we define the Ollivier-Ricci curvature between x and y to be:

$$\kappa(x,y)=\frac{d(x,y)-W(P(x),P(y))}{d(x,y)}. \tag{4}$$

At first glance, it seems strange that this definition could yield a notion of curvature. Again, we refer the reader to [3] for a wealth of intuition. In this work, we content ourselves with a set of suggestive examples:

- An n-dimensional hypercube D_n has positive Ollivier-Ricci curvature for all $x,y\in C_n$.
- Let T be a tree with valence ≥ 3. Then for all $x,y\in T$, $\kappa(x,y)<0$.
- Finally, consider $\mathbb{Z}^n$ with the standard ℓ^1-metric. For any two points x,y with $d(x,y)\geq 2$, $\kappa(x,y)=0$.

These examples draw analogy to the n-sphere, hyperbolic space and Euclidean space, which are the classical examples for positive, negative and zero curvature respectively. In these examples, the sign of curvature is the same for (almost) all x and y. However, we emphasize that curvature is a *local* invariant, and hence is useful for describing heterogeneous geometry. We will shortly see examples of this for the temporal generalization.

Finally, we define the Wasserstein analog of the temporal cost function:

Definition 2.12. Let Γ^t be a temporal network and let $r>0$. Then for all $t\in\mathbb{R}$ and $\mu,\nu\in\mathcal{P}(V)$, we define the r-temporal Wasserstein cost:

$$W_r^t(\mu,\nu)=\inf_{\zeta\in\Pi(\mu,\nu)}\int d_r^t(x,y)d\zeta(x,y). \tag{5}$$

We remark that W_r^t is no longer a metric with d_r^t as a cost functional. However, it achieves its infimal value due to the lower semicontinuity of d_r^t (see [8] for generalities on optimal transport).

Temporal curvature:

With the above definitions and motivation established, we may now define temporal (Ricci) curvature, which is a function $\kappa_r^t : \Gamma \times \Gamma \times \mathbb{R} \to \mathbb{R}$.

Definition 2.13. Let Γ^t be a temporal network. Then the temporal (Ollivier-Ricci) curvature between x and y in (X, ρ^t) at time $t \in \mathbb{R}$ is

$$\kappa_r^t(x,y) = \frac{d_r^t(x,y) - W_r^{t+r}(P(x)^{t+r}, P(y)^{t+r})}{d_r^t(x,y)}. \quad (6)$$

Thus we see that temporal curvature is a simple modification of Ollivier-Ricci curvature with the metric d replaced by the time-varying cost d^t. We remark that, due to the asymmetry of d_r^t, κ_r^t is also asymmetric, i.e. $\kappa_r^t(x,y) \neq \kappa_r^t(y,x)$ in general. To our knowledge, this is the first notion of curvature that can apply to a time-evolving network. We will now demonstrate that this curvature is more than just a convenient adaptation of curvature from the static case to the temporal one, and in fact captures subtle geometric features of temporal networks. For the remainder of this section we will demonstrate this from a theoretical angle, and in the following section we will discuss some applications.

We start with the simple observation that temporal curvature in fact generalizes Ollivier-Ricci curvature. This an almost immediate corollary of Proposition 2.1.

Proposition 2.2. *Let Γ^t be a static temporal network. Then for all $t \in \mathbb{R}$ and $x, y \in V$, $\kappa_r^t(x,y) = \kappa(x,y)$, where κ denotes the Ollivier-Ricci curvature of Γ^{t_0} for any fixed $t_0 \in \mathbb{R}$.*

Next, we introduce a key result about Ollivier-Ricci curvature, and show that the analog holds for the temporal curvature as well. It is well established in the literature that a positive lower bound of $\kappa(x,y) \geq K > 0$ for all $x, y \in \Gamma$ is equivalent to an exponential convergence of the simple random walk on Γ to equilibrium at a rate of $(1-K)^t$ (see Proposition 20 of [3] for a precise statement). This not only provides an invaluable interpretation of what positive Ollivier-Ricci curvature detects in a graph, it also suggests using this curvature as a tool to prove convergence of random walks (more generally Markov chains) in a geometric fashion (see [9] for some particularly striking applications of this fact). We remark that this result was actually discovered earlier by Bubbly and Dyer in their celebrated paper on path coupling [10]. We now state a temporal version of this result:

Theorem 2.1. *Let Γ^t be a temporal network. Then $\kappa_r^t(x,y) \geq K > 0$ for all $x, y \in X$, $t \in \mathbb{R}$ iff for any two probability distributions $\mu, \nu \in \mathcal{P}(X)$, one has*

$$W_r^{t+r}(\mu * P^{t+r}, \nu * P^{t+r}) \leq (1-R) W_r^t(\mu, \nu).$$

This result suggests that positive temporal curvature implies a quantitative form of connectivity for a temporal network, in that information "spreads" quickly over a positively curved temporal network. This fact will provide inspiration for a number of applications in the sequel. We end this section with some examples demonstrating that, despite the above two results relating Ollivier and temporal curvature, the two notions temporal curvature is distinct, in that bounds on the Ollivier-Ricci curvature of the static networks appearing in the sequence of graphs defining Γ^t do not imply bounds on the temporal Ricci curvature, and vice versa. This shows that temporal curvature detects new phenomena that is not explainable in terms of "static invariants", which helps highlight the richness of temporal networks as distinct objects from unrelated sequences of static networks.

Nonnegative static curvature for all nodes at each time, negative temporal curvature between two nodes

Let Γ^t be a temporal network with vertex set $V = \{v_1, v_2, ..., v_8\}$ and edge sets given by:

- For all $k \in \mathbb{N}$ and $t \in [4k, 4k+1]$, $E = \{(v_i, v_{i+1}) | 1 \leq i \leq 7\} \cup \{(v_8, v_1)\}$;
- For all $k \in \mathbb{N}$ and $t \in [4k+1, 4(k+1)]$, $E = \{\{v_{\sigma(i)}, v_{\sigma(i+1)}\} | 1 \leq i \leq 8\}$ where $\sigma \in S_8$ is the permutation defined:

$$\begin{array}{ccc} 1 & \to & 1 \\ 2 & \to & 7 \\ 3 & \to & 6 \\ 4 & \to & 3 \\ 5 & \to & 2 \\ 6 & \to & 4 \\ 7 & \to & 5 \\ 8 & \to & 8 \end{array}$$

Observe that Γ^t is periodic as a function of t, with period 4. At each $t \in \mathbb{R}$, Γ^t is isomorphic to C_8, the cycle graph of length 8, which is a graph with $\kappa(x,y) \geq 0$ for all $x, y \in V$. In particular, $\kappa(v_1, v_5) = \frac{1}{2}$. For any $k \in \mathbb{N}$, $d_1^{4k}(v_1, v_5) = 2$, which is realized by the path that starts at v_1 at time $4k$, then travels to v_5 at time $4k+1$, arriving at $4k+2$. On the other hand, one may compute $W_1^{4k+1}(P(v_1)^{4k+1}, P(v_5)^{4k+1}) = \frac{7}{3}$, and hence $\kappa_1^{4k+1}(P(v_1)^{4k+1}, P(v_5)^{4k+1}) = -\frac{1}{6}$. Hence the temporal curvature from v_1 to v_5 is negative despite the Ollivier-Ricci curvature between these nodes being positive at every time step. If one constructs a similar example on C_n for any even $n \in \mathbb{N}, n \geq 8$ with period $n/2$, one will see the same behavior, and furthermore will find that the minimum value of κ_1^{nk+1} can become arbitrarily negative as n increases. From this example we conclude:

Lemma 2.1. *Nonnegative (positive) static curvature at each $t \in \mathbb{R}$ does not imply nonnegative (positive) temporal curvature for a temporal network Γ^t.*

Positive temporal curvature at all times for (most) nodes, zero static curvature for all times at all nodes

Let Γ^t be a temporal network on the vertex set $V = \{(i,j) | i,j \in \mathbb{Z}\}$ and edge set $E = \{(i,j), (i \pm 1, j) | i,j \in \mathbb{Z}\} \cup \{(i,j), (i, j \pm 1) | i,j \in \mathbb{Z}\}$. In other words Γ^t is isomorphic to $\mathbb{Z}^2$. We equip every edge in E with a function $w^t(e) = 1$ for all $t \leq 1$ and $w^t(e) = \frac{1}{t}$ for $t > 1$. The data (V, E, w^t) define a temporal metric space, where at a fixed time t_0, ρ^t is defined as the shortest path distance in the weighted graph (V, E, w^{t_0}), and one may compute that $\kappa(x,y) = 0$ for all $x, y \in V$. Let κ^{4k+1} denote the temporal curvature w.r.t. the uniform kernel P^t of radius 1 on Γ^t, with communication parameters $(1,1)$. Hence for $k \in \mathbb{Z}_{\geq 0}$, $P^k(x)$ is the uniform measure supported on $\{(i,j) \in \mathbb{Z}^2 || i|, |j| \leq k\}$. From the definition, $d^k(x,y) \geq 1$

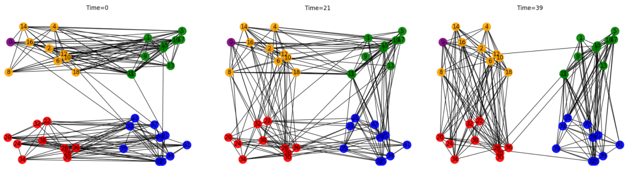


Figure 1: Example static networks $\Gamma^0, \Gamma^{21}, \Gamma^{39}$ sampled from the random temporal network Γ^t, color coded by temporal community: Green nodes denote community $V_{0,0}$, yellow nodes denote community $V_{1,0}$, blue nodes denote community $V_{0,1}$ and red nodes denote community $V_{1,1}$. The purple node denotes the node labelled 0.

for all $x \neq y$. However, one may also verify that for x, y with the metric $\rho^k(x,y) > 1$, $d_1^{k+1}(x,y) < d_1^k(x,y)$. For instance, if $x = (x_1, x_2)$ and $y = (x_1 + 2, x_2)$, then $d_1^0(x,y) = 2$, which is realized by the path that waits at x on the interval $[0,1)$ then travels to y at time 1, arriving at time 2, whereas $d_1^1(x,y) = 1$, which is realized by the path that immediately travels to y at time 1, arriving at time 2.

For any $x, y \in \mathbb{Z}^2$, there is a transport plan $\zeta_{x,y}^k$ between $P^k(x)$ and $P^k(y)$ that maps the mass at $x + z$ to $y + z$ for all $z \in \{(i,j) \in \mathbb{Z}^2 | |i|, |j| \leq k\}$, and the cost of this plan is equal to $d^k(x,y)$ via symmetry considerations. This provides the upper bound $W_1^k(P(x)^k, P(y)^k) \leq d_1^k(x,y)$ for all x, y, and hence $W_1^{t+k}(P(x)^k, P(y)^k) \leq d_1^{k+1}(x,y) < d_1^k(x,y)$ for any x, y with $\rho^k(x,y) > 1$. From this example we conclude:

Lemma 2.2. *There exist infinite temporal graphs Γ^t with nonnegative temporal curvature and strictly positive temporal curvature outside of a finite set at all times $t \in \mathbb{R}$, but everywhere zero Ollivier curvature at each time t. Hence zero Ollivier-Ricci curvature everywhere for each network in Γ^t does not imply zero temporal curvature for Γ^t as a temporal network.*

Application: Community detection in time-evolving networks

In the theory of (static) networks, Ollivier-Ricci curvature has been used for community detection applications, see for instance [4]. The idea is that one can subdivide a network into communities such that within a community Ricci curvature is positive and between communities it is negative. Due to results such as Theorem 2.1, one may interpret this as subdividing the network into regions such that within a region information flow occurs rapidly and between regions information "bottlenecks". Another related idea was explored in [11], where Ollivier-Ricci flow was applied to a network to isolate positively curved regions, which are taken to be communities. In the temporal setting, community detection becomes a much more subtle problem than in the static case. Not only does one need to potentially track communities that change over time, but many of the standard methods one would think to apply to graphs do not apply directly, such as the spectral methods pursued in [12] (though see [13] where spectral methods are adapted to the temporal setting).

In this section we investigate how the ideas in [4] may be extended to community detection in time-evolving networks using temporal curvature. In particular, we show that temporal curvature may detect the formation and evolution of time evolving communities.

To demonstrate this, we define the following random temporal network Γ^t: Set $|V| = 40$, and label these nodes with the numbers $\{0, 1, ..., 39\}$. Define two partitions I and J of the set $\{0, 1, ..., 39\}$, where partition I has parts $I_1 = \{x \leq 19 | x \in \{0, 1, ..., 39\}\}$ and $I_2 = \{x > 10 | x \in \{0, 1, ..., 39\}\}$, and partition J has parts $J_1 = \{x(\mod 2) = 0 | x \in \{0, 1, ..., 39\}\}$ and $J_2 = \{x(\mod 2) = 1 \leq 19 | x \in \{0, 1, ..., 39\}\}$. We let A_I denote the $(0.5, 0.01)$-block matrix with blocks I and A_J denote the $(0.5, 0.01)$-block matrix with blocks J. For any $t \in [0,1]$, let $A^t = tA_I + (1-t)A_J$, and define the family $A(k) := \{A^t | t = \frac{k}{20}, k \in \{0, 1, ..., 19\}\}$. We then sample 10 random networks from Γ_{A_I}, a single random network from $A(k)$ for each k, and then 10 random networks from Γ_{A_J}. We enumerate this sequence of networks $\Gamma_1, \Gamma_2, ..., \Gamma_{40}$ and assign the i'th network interval $[i, i+1]$. We realize each Γ_i as a constant temporal network on the interval $[i, i+1]$, and concatenate these networks to obtain a (stochastic) temporal network Γ defined on the interval $[0, 40]$. In Figure 1, we display some slices from a sample from the random network Γ^t, which demonstrates the evolution of the community structure.

We fix the node labelled 0, and observe that nodes in (a sample from) Γ^t fall into four temporal communities defined relative to 0. Denote $p_{0,w}^t$ for the probability that an edge exists between 0 and w at time t, and define the following subsets (called '*communities*') of $\{0, 1, ..., 39\}$:

- $V_{0,0} = \{w \in V | \{w \leq 19\} \cap \{w \pmod 2 = 0\}\}$;
- $V_{1,0} = \{w \in V | \{w \leq 19\} \cap \{w \pmod 2 = 1\}\}$;
- $V_{0,1} = \{w \in V | \{w > 19\} \cap \{w \pmod 2 = 0\}\}$;
- $V_{1,1} = \{w \in V | \{w > 19\} \cap \{w \pmod 2 = 1\}\}$.

Nodes in $V_{0,0}$ always have a high probability of being neighbors with 0, whereas nodes in $V_{1,1}$ always have a low probability of being neighbors with 0. Nodes in $V_{1,0}$ (resp. $V_{0,1}$) initially have a high probability (resp. low probability) of being neighbors with v which then decreases (resp. increases) as a function of time. These differences motivate us to regard $V_{0,0}, V_{0,1}, V_{1,0}, V_{1,1}$ as four different temporal communities in Γ^t.

For a sample from Γ^t, we compute $\kappa_r^t(0, w)$ for all $w \neq 0 \in V$ and $t \in \{0, 1, ..., 39\}$. Then for each class $V_{a,b}$, we compute $\frac{1}{|V_{a,b}|} \sum_{w \in V_{a,b}} \kappa_1^t(0, w)$ for all t and plot the results in Figure 2. We see that the intuition from the static case, as established in [4], holds for the temporal curvature as well, in that a high positive (averaged) temporal curvature tracks detects when nodes are likely to be in the same community as the node 0, and conversely lower (averaged) curvature detects when nodes are in a different community than 0. Interestingly, the averaged curvatures are all approximately equal for $t \in [15, 25]$, which indicates the relatively homogeneous structure of the network at this time (see for example the middle network in Figure 1). These preliminary results suggest that temporal curvature may indeed find usage as a clustering tool in temporal networks.

3. Persistence and Time-Varying Networks

Zigzag Persistence

In this paper, we will introduce and utilize the theory of zigzag persistence to study time-varying networks. In

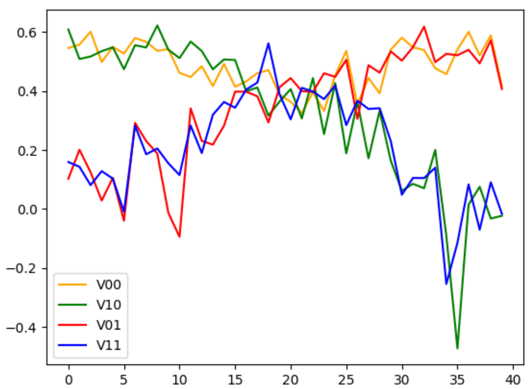


Figure 2: Curvatures $\kappa(0,w)^t$ averaged over the communities $V_{0,0}, V_{0,1}, V_{1,0}, V_{1,1}$ as a function of time.

[14], a time-varying network is defined to be a sequence $G_1, G_2, \ldots, G_n$ of graphs, with $G_i = (V_i, E_i)$. For simplicity, we will follow the convention in [14] and specify that the vertex set remains constant, meaning $V_1 = \cdots = V_n$. We will assume there is always an inclusion map $G_i \leftrightarrow G_{i+1}$, where $\leftrightarrow$ indicates that the inclusion can go in either direction. Indeed, it is fair to assume that we have these inclusions: this amounts to assuming that one or more communication links are dropped at the same time, or one or more communication links are formed at the same time. However, we cannot have a mixture of communication links entering and leaving the network at the same time. With this assumption, what we get in return is a zigzag sequence, as introduced in [15]:

$$G_1 \leftrightarrow G_2 \leftrightarrow \cdots \leftrightarrow G_n.$$

We will adopt terminology from [15] to discuss the type of a zigzag sequence. Denote a left arrow with a g, $X_i \xleftarrow{g_i} X_{i+1}$, and denote a right arrow with a f, $X_i \xrightarrow{f_i} X_{i+1}$. The *type* of a zigzag sequence is a *word* consisting of f, g that denotes the order and direction of all maps in the sequence. For example, the zigzag sequence $X_1 \leftarrow X_2 \rightarrow X_3$ has type gf.

In [14], the authors propose using zigzag persistence to study the structure delay-tolerant time-varying networks and provide an example. We expand on this idea, using zigzag persistence to not only study the structure of time-varying networks, but to also investigate other persistent features. For one, we will answer the question: what paths on our graph persist from time a to time b? In order to do so, we will utilize sheaf theory.

Sheaves on Graphs

It is already known that sheaves are a useful tool for networking [2, 16–18], We will utilize more machinery from applied sheaf theory to study the persistence of paths in a time-varying network.

A sheaf on a graph $G = (V, E)$ is an example of a *cellular sheaf*. Consider a sheaf $\mathcal{F}$ valued in a category $\mathcal{C}$. The data of $\mathcal{F}$ consists of objects $\mathcal{F}(v)$ and $\mathcal{F}(e)$ associated to every vertex v and edge e and a morphism $\mathcal{F}(v \leq e) = \mathcal{F}(v) \to \mathcal{F}(e)$ for every incidence relation $v \leq e$.

Now, suppose $f : G_i \to G_{i+1}$ is an inclusion of graphs, and further that $\mathcal{F}_i$ is a sheaf over G_i and $\mathcal{F}_{i+1}$ is a sheaf over G_{i+1}, both valued in the same category. Then, we may define an f-map between $\mathcal{F}_i$ and $\mathcal{F}_{i+1}$ as a natural transformation $\varphi : f_*\mathcal{F}_i \to \mathcal{F}_{i+1}$ of sheaves on G_{i+1}, where the components of φ are given by $\varphi_v = f_*\mathcal{F}_i(v) = \mathcal{F}_i\left(f^{-1}(v)\right) \to \mathcal{F}_{i+1}(v)$. The same definition holds for edges. This gives the naturality square

$$\begin{tikzcd} f_*\mathcal{F}_i(v) \arrow[r, "\varphi_v"] \arrow[d] & \mathcal{F}_{i+1}(v) \arrow[d] \\ f_*\mathcal{F}_i(e) \arrow[r, "\varphi_e"] & \mathcal{F}_{i+1}(e) \end{tikzcd}$$

In this section, we are particularly interested in studying the *path sheaf*, as introduced in [16]. Let $G = (V, E)$ be a graph with both a distinguished source v_S and target v_T vertex. The set-valued path sheaf $\mathcal{P}_G$ on G is defined as follows. For $v \in V$ a vertex, denote by $\mathrm{In(v)}$ the set of edges flowing into v and by $\mathrm{Out(v)}$ the set of edges flowing out of v. Let $\perp$ indicate that a vertex or edge is "off", and let $\top$ indicate that a vertex or edge is "on". Then, on a vertex v, $\mathcal{P}_G(v)$ is defined as follows:

$$\mathcal{P}_G(v) = \begin{cases} \mathrm{Out(v)} & \text{if } v = v_S, \\ \mathrm{In(v)} & \text{if } v = v_T, \\ (\mathrm{Out(v)} \times \mathrm{In(v)}) \cup \{\perp\} & \text{else.} \end{cases}$$

That is, the stalk $\mathcal{P}_G(v)$ tells us the possible "flows" through v (i.e., which edges we enter and leave v through), or it tells us that v is turned "off". On an edge $e \in E$, $\mathcal{P}_G(e)$ is defined as follows:

$$\mathcal{P}_G(e) = \{\perp, \top\},$$

telling us than an edge is either on or off.

A global section of the path sheaf gives us a path from v_S to v_T (or, the disjoint union of such a path with a cycle which is disjoint from the source and target) [2]. The aim of this section is to extend that result to account for time-varying networks, thus identifying persistent paths.

To do so, suppose $f : G_i \to G_{i+1}$ is an inclusion of graphs. Consider an f-map between $\mathcal{P}_i$ and $\mathcal{P}_{i+1}$. The components of the natural transformation $\varphi : f_*\mathcal{P}_i \to \mathcal{P}_{i+1}$ are given by set inclusions. Indeed, for $e \in E_{i+1}$, $\varphi_e : f_*\mathcal{P}_i(e) \to \mathcal{P}_{i+1}(e)$ is given by the inclusion $\emptyset \to \{\perp, \top\}$ if $e \notin E_i$ and the identity map if $e \in E_i$.

For $v = v_S$,

$$\begin{aligned} \varphi_{v_S} :& f_*\mathcal{P}_i(v_S) \to \mathcal{P}_{i+1}(v_S) \\ & \mathrm{Out(v_S)}|_{\mathrm{G_i}} \to \mathrm{Out(v_S)}, \end{aligned}$$

where $\mathrm{Out(v_S)}|_{\mathrm{G_i}}$ indicates the edges in $\mathrm{Out(v_S)}$ that are also in E_i. Define φ_{v_T} similarly.

Finally, for $v \neq v_S, v_T$,

$$\begin{aligned} \varphi_v :& f_*\mathcal{P}_i(v) \to \mathcal{P}_{i+1}(v) \\ & (\mathrm{Out(v)} \times \mathrm{In(v)})|_{\mathrm{G_i}} \cup \{\perp\} \to (\mathrm{Out(v)} \times \mathrm{In(v)}) \cup \{\perp\}. \end{aligned}$$

If $v \notin V_i$, then $f_*\mathcal{P}_i(v) = \emptyset$. In summary, f-maps between path sheaves are induced by inclusions on the underlying graphs.

Now, we want to endow the set-valued path sheaf with the structure of a vector space. We define the vector space-valued

path sheaf in almost the identical way, but apply the *free functor* on every set. The free functor $\mathcal{F}$ turns a set S into the vector space $\mathbb{F}(S)$ over the field $\mathbb{F}$ generated by the elements of S. For purposes that will become clear in future proofs, we will declare that $\perp = 0$ in our vector space. We will denote the vector space-valued path sheaf on the graph G_i by $\mathcal{P}_i^{\mathbb{F}}$.

This vector space structure will allow us to study dynamics on our sheaf through the *sheaf Laplacian* [19]. Indeed, the kernel of the sheaf Laplacian of the vectorized path sheaf gives global sections (paths from source to target) of the sheaf [2]. We aim to study the persistence of such paths.

Persistent Path Sheaf Laplacians for Zigzag Sequences

In this section, we will build upon work in [20], where the authors generalize the sheaf Laplacian to the *persistent* sheaf Laplacian utilizing previous work from [7, 19]. Just like the kernel of the sheaf Laplacian gives us information about paths from v_S to v_T, we will use the persistent sheaf Laplacian to find paths from v_S to v_T that persist on a given time interval.

Consider a zigzag sequence

$$(G_1, \mathcal{P}_1^{\mathbb{F}}) \leftrightarrow (G_2, \mathcal{P}_2^{\mathbb{F}}) \leftrightarrow \cdots \leftrightarrow (G_n, \mathcal{P}_n^{\mathbb{F}})$$

consisting of networks $G_1, \ldots, G_n$ with their corresponding path sheaves $\mathcal{P}_1^{\mathbb{F}}, \ldots, \mathcal{P}_n^{\mathbb{F}}$. The morphisms $(G_i, \mathcal{P}_i^{\mathbb{F}}) \leftrightarrow (G_{i+1}, \mathcal{P}_{i+1}^{\mathbb{F}})$ consist of graph inclusions $G_i \leftrightarrow G_{i+1}$ and f- or g-maps $\mathcal{P}_i^{\mathbb{F}} \leftrightarrow \mathcal{P}_{i+1}^{\mathbb{F}}$.

Here, we will generalize the persistent sheaf Laplacian from [20] so that it may be applied to the path sheaf on time-varying networks. Our first step is to define projection maps between sheaf cochain spaces. This algebra will ensure that we can generalize the work from [20] to f-maps between path sheaves. Let

$$C^k(G_i; \mathcal{P}_i^{\mathbb{F}}) = \bigoplus_{\sigma \in G_i \,|\, \dim \sigma = k} \mathcal{P}_i^{\mathbb{F}}(\sigma)$$

be the direct sum of the vector spaces $\mathcal{P}_i^{\mathbb{F}}(\sigma)$ for all σ of dimension k in G_i. For our purposes, we are only interested in $k = 0, 1$ (vertices and edges). We may define the coboundary operator $d_i : C^0(G_i, \mathcal{P}_i^{\mathbb{F}}) \to C^1(G_i, \mathcal{P}_i^{\mathbb{F}})$ on a vertex v and extend linearly:

$$d_i(v) = \sum_{e \,|\, v \leq e} [v : e] \mathcal{P}_i^{\mathbb{F}}(v \leq e),$$

where $[v : e] = 1$ if v is the head of e, and $[v : e] = -1$ if v is the tail of e. Using this coboundary operator, we obtain a cochain complex:

$$0 \to C^0(G_i, \mathcal{P}_i^{\mathbb{F}}) \xrightarrow{d_i} C^1(G_i, \mathcal{P}_i^{\mathbb{F}}) \to 0.$$

Now, suppose there is an inclusion $f : G_i \to G_{i+1}$ and an f-map $\mathcal{P}_i^{\mathbb{F}} \to \mathcal{P}_{i+1}^{\mathbb{F}}$. Define maps

$$\begin{aligned} f^0 :& C^0(G_{i+1}, \mathcal{P}_{i+1}^{\mathbb{F}}) \to C^0(G_i, \mathcal{P}_i^{\mathbb{F}}), \text{ and} \\ f^1 :& C^1(G_{i+1}, \mathcal{P}_{i+1}^{\mathbb{F}}) \to C^1(G_i, \mathcal{P}_i^{\mathbb{F}}) \end{aligned}$$

on basis elements and extend linearly. Let $s \in \mathcal{P}_{i+1}^{\mathbb{F}}(v)$. Define

$$f^0(s) = \begin{cases} s & \text{if } s \in \mathcal{P}_i^{\mathbb{F}}(f^{-1}(v)) \\ 0 & \text{if } s \notin \mathcal{P}_i^{\mathbb{F}}(f^{-1}(v)) \end{cases}.$$

Define f^1 in the exact same way.

Proposition 3.1. *The sequence of f^ks defines a cochain map. That is, $df^0 = f^1 d$.*

$$\begin{tikzcd} 0 \arrow[r] & C^0(G_i, \mathcal{P}_i^{\mathbb{F}}) \arrow[r, "d_i"] & C^1(G_i, \mathcal{P}_i^{\mathbb{F}}) \arrow[r] & 0 \\ 0 \arrow[r] & C^0(G_{i+1}, \mathcal{P}_{i+1}^{\mathbb{F}}) \arrow[r, "d_{i+1}"] \arrow[u, "f^0"] & C^1(G_{i+1}, \mathcal{P}_{i+1}^{\mathbb{F}}) \arrow[r] \arrow[u, "f^1"] & 0 \end{tikzcd}$$

A proof of this proposition can be found in the appendix. This proof does rely heavily on the nice structure of the path sheaf, making our generalizations possible. This proposition tells us that a single inclusion of networks $f : G_i \to G_{i+1}$ and the f-map on their path sheaves $\mathcal{P}_i^{\mathbb{F}}, \mathcal{P}_{i+1}^{\mathbb{F}}$ induces a cochain map. Now, we will generalize to zigzag sequences.

Consider a time-varying graph with the path sheaf at each step

$$(G_1, \mathcal{P}_1^{\mathbb{F}}) \leftrightarrow (G_2, \mathcal{P}_2^{\mathbb{F}}) \leftrightarrow \cdots \leftrightarrow (G_n, \mathcal{P}_n^{\mathbb{F}}).$$

Using the tools above, we get a sequence of cochain maps

$$\begin{tikzcd} 0 \arrow[r] & C^0(G_1, \mathcal{P}_1^{\mathbb{F}}) \arrow[r, "d_1"] \arrow[d, leftrightarrow] & C^1(G_1, \mathcal{P}_1^{\mathbb{F}}) \arrow[r] \arrow[d, leftrightarrow] & 0 \\ 0 \arrow[r] & C^0(G_2, \mathcal{P}_2^{\mathbb{F}}) \arrow[r, "d_2"] \arrow[d, leftrightarrow] & C^1(G_2, \mathcal{P}_2^{\mathbb{F}}) \arrow[r] \arrow[d, leftrightarrow] & 0 \\ & \cdots \arrow[d, leftrightarrow] & \cdots \arrow[d, leftrightarrow] & \\ 0 \arrow[r] & C^0(G_n, \mathcal{P}_n^{\mathbb{F}}) \arrow[r, "d_n"] & C^1(G_n, \mathcal{P}_n^{\mathbb{F}}) \arrow[r] & 0 \end{tikzcd}$$

Now, to get a map $p : C^k(G_n, \mathcal{P}_n^{\mathbb{F}}) \to C^k(G_1, \mathcal{P}_1^{\mathbb{F}})$, we compose induced maps on right arrows f^k and adjoints of induced maps on left arrows $(g^k)^*$.

Finally, we may define the *persistent sheaf Laplacian* for the path sheaf of a time-varying network as in [20].

Let $p : C^0(G_j, \mathcal{P}_j^{\mathbb{F}}) \to C^0(G_i, \mathcal{P}_i^{\mathbb{F}})$ be as above. Define

$$C^{i,j} = \{e \in C^1(G_j, , \mathcal{P}_j^{\mathbb{F}}) \mid d_j^*(e) \in C^0(G_i, , \mathcal{P}_i^{\mathbb{F}})\}.$$

This space may be interpreted as edges (i.e., communication links) at time j whose boundary incident nodes exist at time i. Now, define

$$\partial^{i,j} = \left(p(d_j^*|_{C^{i,j}})\right)^*.$$

This data is summarized in the following diagram.

$$\begin{tikzcd} C^0(G_i, \mathcal{P}_i^{\mathbb{F}}) \arrow[rr, "d_i"] & & C^1(G_i, \mathcal{P}_i^{\mathbb{F}}) \\ & C^{i,j} \arrow[ul, "\partial^{i,j}"'] \arrow[dr, dashed] & \\ C^0(G_j, \mathcal{P}_j^{\mathbb{F}}) \arrow[uu, "p"] \arrow[rr, "d_j"] & & C^1(G_j, \mathcal{P}_j^{\mathbb{F}}) \arrow[uu, "p"'] \end{tikzcd}$$

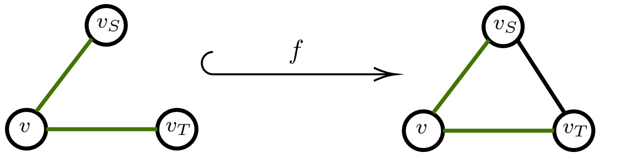


Figure 3: A time-varying graph with an inclusion $G_1 \to G_2$. The persistent path from v_S to v_T is shown in green.

Define the persistent sheaf Laplacian $L_k^{i,j}$ (for $k = 0, 1$) as:

$$L_0^{i,j} = \left(\partial^{i,j}\right)^* \partial^{i,j}$$
$$L_1^{i,j} = d_i(d_i)^*.$$

We will primarily be interested in $L_0^{i,j}$, which will give us information about paths that persistent from time i to time j.

Example 3.1. See Figure 3 for a small example of a time-varying graph. We want to find which paths (from the source to the target) persist. The persistent Lapalcian is given by the matrix

$$L_0^{1,2} = \begin{pmatrix} 1 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 1 \end{pmatrix},$$

which has kernel

$$\ker L_0^{1,2} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}.$$

This verifies that the highlighted green path is, indeed, the only path that persists from time 1 to time 2.

Although this tool is useful for our purposes, it is highly depended on the nice structure of f-maps between path sheaves. Indeed, the commutativity of the diagram in Proposition 3.1 is not guaranteed to work for general zigzag sequences with arbitrary sheaves and f-maps. Further, this method does not address one fallback of the vectorized path sheaf: as shown in [2], there are examples of global sections of the vector space-valued path sheaf that are not paths in the underlying graph. Thus, we have reason to believe the same problem occurs when identifying persistent paths.

4. Network Embeddings

A **network embedding** is a low-dimensional representation of a graph in a vector space that preserves the global network topology, or an embedding map $\Phi : V \to \mathbb{R}^n$ which assigns each node $v \in V$ to a corresponding vector in Euclidean space. Embedding the nodes in a low-dimensional feature space helps reduce the complexity of network analysis algorithms such as node classification [21], link prediction [22], and community detection [11].

In this section, we consider and analyze several methods of network embedding and their geometric properties. Additionally, we investigate the effects of embedding time-varying networks in latent spaces according to their graph curvature at each time step. The resulting sequence of embeddings (point clouds) in low-dimensional spaces can then be used as input for a diverse range of data filtrations used in TDA [23]. We speculate a correspondence between the embedding method and the type of filtration applied to the embedded point cloud. For instance, a Vietoris-Rips filtration may be more suited for Euclidean embeddings, whereas a k-clique filtration may be better capture the geometry of hyperbolic embeddings.

Consider a snapshot representation of a temporal network, defined as a sequence of graphs $\{G_t\}_{t\geq 0}$ indexed by time $t \in \mathbb{R}$. For each $t \in \mathbb{R}$, we want to compute a node embedding map $\Phi_t : V_t \to \mathbb{R}^d$ of the graph $G_t = (V_t, E_t)$ into a vector space of a fixed dimension d (future work may integrate time-dependent dimensions d_t). We will obtain a sequence of point clouds $\{\Phi_1(V_1), \Phi_2(V_2), ..., \Phi_n(V_n)\}$ embedded in a target vector space. An important issue of learning temporal embeddings is the problem of *alignment*: the learned vectors may not be placed in the same latent space over time. To solve this, embeddings can be learned independently for each time step using static methods and an optimal linear transformation of the output matrices minimizing the distance between consecutive embeddings can be computed [24]. Other approaches learn current time step representations by initializing them with previous embedding vectors [25,26].

Spectral Embeddings

The objective of a spectral embedding is to compute an embedding matrix of size $\mathbb{R}^{|V|\times d}$ by exploiting the spectral properties of the graph Laplacian. One such embedding method is **Laplacian Eigenmaps** (LE) [27]. The LE embedding of a graph in dimension k is computed by solving for the matrix of eigenvectors corresponding to the k smallest nonzero eigenvalues of the generalized eigenvalue problem $\mathbf{Ly} = \lambda\mathbf{Dy}$, where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian, D is the diagonal matrix of vertex degrees, and W is the edge weight matrix. Refer to [27] for the detailed algorithm.

Laplace-Beltrami Operator—The LE embedding algorithm assumes the network was constructed as a similarity graph from data points lying on a manifold. There is a direct correspondence between the graph Laplacian and the Laplace-Beltrami operator $\mathcal{L}$ on the underlying manifold, defined by $\mathcal{L}(f) = -\operatorname{div}\nabla(f)$. Suppose $\mathcal{M}$ is a smooth, compact m-dimensional Riemannian manifold embedded in $\mathbb{R}^l$, where the Riemannian metric on $\mathcal{M}$ is induced by the metric on $\mathbb{R}^l$. Let $f : \mathcal{M} \to \mathbb{R}$ be a scalar function on the manifold. Then, the gradient ∇f is a vector field on the manifold and its norm $\|\nabla f\|$ gives an estimate of how far apart f maps nearby points. We look for a map that preserves locality by minimizing $\int \|\nabla f(x)\|^2$ on $\mathcal{M}$. The connection to the graph Laplacian is that this corresponds to minimizing $L\mathbf{f} = \mathbf{\frac{1}{2}\sum_{i,j}(f_i - f_j)^2 W_{ij}}$ on a graph. By Stokes' Theorem, since $\int_{\mathcal{M}}\langle \mathbf{X}, \nabla\mathbf{f}\rangle = -\int_{\mathcal{M}} \operatorname{div}(\boldsymbol{X})\mathbf{f}$ for a vector field $\boldsymbol{X}$, we have that

$$\int_{\mathcal{M}} \|\nabla f\|^2 = \int_{\mathcal{M}} \mathcal{L}(f)f \tag{7}$$

where $\mathcal{L}$ has a discrete spectrum $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \ldots$, and f_i are the eigenfunctions corresponding to eigenvalues λ_i. The k-dimensional embedding is given by $\mathbf{x} \to (\mathbf{f_2}(\mathbf{x}), \ldots, \mathbf{f_{k+1}}(\mathbf{x}))$.

A slightly modified version of LE is the embedding method **Geometric Laplacian Eigenmaps** (GLEE) [28]. GLEE is a geometric approach to network embedding that is motivated by the correlation between the geometric properties

of a simplex and the spectral properties of a graph [29]. Formally, given a graph $\mathcal{G}$, we first compute the singular value decomposition (SVD) of its Laplacian $\mathbf{L} = \mathbf{S}\mathbf{S}^\top$, where $\mathbf{S}$ is a unique $n \times n$ matrix. As the graph Laplacian is a symmetric positive semi-definite matrix, its singular values coincide with its eigenvalues. If we let $\mathbf{S} = \mathbf{P}\sqrt{\mathbf{\Lambda}}$, then the SVD of L is simply the spectral decomposition $\mathbf{L} = \mathbf{P\Lambda P}^\top$, where P is the matrix with columns as the eigenvectors of L and Λ is the diagonal matrix of corresponding eigenvalues. The k-dimensional vector embedding of node i is given by the i-th row of $\mathbf{S^k}$, the matrix of the first k columns of $\mathbf{S}$. In fact, the embedding matrix $\mathbf{S^k}$ is the matrix of rank k that is closest to L in Frobenius norm. That is,

$$\left\|\mathbf{L} - \mathbf{S^k}\left(\mathbf{S^k}\right)^\top\right\|_F \leq \|\mathbf{L} - \mathbf{M}\|_F \tag{8}$$

for any other rank k matrix M.

Simplex Geometry—Given a set of $k+1$ affinely independent points $\{p_i\}_{i=0}^k$ in $\mathbb{R}^k$, their convex hull is called a **simplex**. As Fiedler proved in [30], every undirected connected graph $\mathcal{G}$ with n nodes corresponds to a unique $(n-1)$-dimensional simplex that perfectly encodes the structure of $\mathcal{G}$. That is, every geometric invariant of the simplex is also a graph invariant. To see this, suppose $\mathcal{G}$ is a connected graph with n nodes, and consider its (positive semi-definite) Laplacian matrix $\mathbf{L}$ with rank $n-1$. As stated in [28], for a positive semi-definite $k \times k$ matrix $\mathbf{Q}$ with decomposition $\mathbf{Q} = \mathbf{S}\mathbf{S}^\top$, the rows of $\mathbf{S}$ lie at the vertices of a simplex if and only if $\text{rank}(\mathbf{Q}) = \mathbf{k-1}$. Letting $\mathbf{S} = \mathbf{P}\sqrt{\mathbf{\Lambda}}$, we see that $\mathbf{L} = \mathbf{S}\mathbf{S}^\top$ and the rows of $\mathbf{S}$ form the vertices of a $(n-1)$-dimensional simplex. This is denoted the *simplex of* $\mathcal{G}$. We can recover the degree and adjacency information of node i by computing the lengths and dot products of the simplex vertices:

$$\deg(i) = \|s_i\|^2 \quad , \quad \mathrm{a_{ij}} = -\mathrm{s_i} \cdot \mathrm{s_j^\top}. \tag{9}$$

Further, we can compute the number of common neighbors (CN) between two nodes in terms of the centroids of node sets:

$$CN(i,j) = -\deg(i)\, C_{N(i)} \cdot s_j^\top, \tag{10}$$

where $N(i)$ is the neighborhood of node i, and $C_{N(i)} = \frac{1}{|N(i)|}\sum_{i \in N(i)} s_i$ is the centroid of the node set $N(i)$.

Temporal Network Embeddings for Link Prediction

We can extend the aforementioned static embeddings to temporal embeddings in several ways. The simplest method is to aggregate all of the adjacency matrices A^t into a single adjacency matrix $A_{\text{aggregate}}[i][j] = \sum_{t=1}^T \lambda^{T-t} A^t[i][j]$ with weight $\lambda \in (0,1)$, and apply a static graph embedding technique [31, 32]. Alternatively, we can learn snapshot representations jointly by defining a temporal similarity matrix that captures frequencies by which vertices co-appear in the set of random walks over graph vertices [33]. This is the method of **TemporalNode2vec**, a temporal network embedding approach based on Node2vec [34]. The similarity matrix (PPMI) is defined as follows:

$$\mathrm{PPMI}_t(v_1, v_2) = \max\left(0, \log\left(\theta \frac{|v_1, v_2|_t^w \cdot |V|}{|v_1|_t \cdot |v_2|_t}\right)\right)$$

where $|v_1, v_2|_t^w$ is the number of times v_1 and v_2 co-appear in the set of walks of G_t within a window of size w, $|v_i|_t$ is the number of occurrences of v_i in the set of walks of G_t, and θ is a tunable hyper parameter. We then seek to minimize

$$\sum_{t=1}^T \left\|PPMI_t - U_t U_t^T\right\|_F^2$$

where U_t are the embedding matrices consisting of learned vectors $u_i(t)$ satisfying $u_1(t)^T u_2(t) \approx PPMI_t(v_1, v_2)$. Depending on the geometry of our network, we can compute embeddings of networks in non-Euclidean spaces, including hyperbolic spaces or Riemannian manifolds [35]. For example, if the Ollivier-Ricci curvature of the network is negative, then it may be better to embed the network in hyperbolic space. Whereas if the curvature is positive, then we should embed the graph in a latent space (or Riemannian manifold) of positive Ricci curvature. A recently developed hyperbolic temporal network embedding algorithm has proven to more accurately capture the hierarchical (tree-like) structure of networks [36]. For hyperbolic geometry, there are two mathematically equivalent models for latent space representations: the Poincaré ball model and the Lorentz model [35]. A network is **congruent** with its underlying latent geometry if the *shortest* paths correspond directly to *geodesic* paths in its latent space representation.

5. Lattice Sheaves

Recently, applications of sheaf theory to networking has seen increasing attention. For instance in [16], the authors demonstrate that paths from a fixed source to target in a directed graph can be encoded by the set of global sections of a particular sheaf. More recently, cellular sheaves of vector spaces, such as in Section 3, were used to study the same routing problem using powerful tools from homological algebra. Within that framework, vector addition does not capture the superposition of two paths, and consequently, some global sections do not correspond to realizable paths in the network. In this section, we explore other data structures, such as lattices, for modelling routing problems on a directed network. One advantage to this perspective is that ideas from fuzzy logic can be used to encode certain network properties, such as temporal structure.

A **lattice** is a partially ordered set with a greatest lower bound operation $\wedge$ called the **meet** and a least upper bound operation $\vee$ called the **join**. Throughout this section we will be primarily interested in **bounded** lattices, which have a least element $\perp$ and a greatest element $\top$. A morphism between two lattices L and K are **join preserving** functions $g : L \to K$, i.e. $g(a \vee b) = g(a) \vee g(b)$ and $g(\perp_L) = \perp_K$.

Given any set X, the power set lattice 2^X has a partial order given by set inclusion. The meet and join operations are given by intersection and union, respectively. Morphisms between lattices 2^X and 2^Y can be represented by binary relations $R \in 2^{X \times Y}$ where R acts on a subset $A \subseteq X$ by

$$R(A) = \{y \in Y \mid \exists x \in A,\ R(x,y) = 1\}.$$

Note that every set function $f : X \to Y$ can be viewed as a relation in $2^{X \times Y}$. We now recall some useful characteristics of lattices.

Definition 5.1. Let L be a bounded lattice. We say that

- $b \in L$ **covers** an element a, denoted $a <: b$, if $a < b$ and there is no element c such that $a < c < b$. An element that covers $\perp$ is called an **atom**.

- L is **atomic** if for every $b \in L$, there exists an atom $a \leq b$.

- L is **atomistic** if every element $\ell > \perp$ can be expressed as the join of atoms.

- L is **semi-modular** if $a \wedge b <: a$ implies $b <: a \vee b$.

- L is a **geometric lattice** if it is finite, atomistic, and semi-modular.

Fuzzy Sets and Logic

Fuzzy logic generalizes the notion of truth-values beyond the Boolean lattice to any complete lattice L. Fuzzy values can be used to encode the degree to which a statement is true or encode more complex relationships between logic statements. For our discussion, we will use a special type of lattice which has a rich algebraic theory.

Definition 5.2. A bounded lattice $L = (L, \perp, \top, \wedge, \vee)$ is said to be **residuated** it has two additional binary operations $\oplus$ and $\rightarrow$ satisfying the following properties.

1. $(L, \oplus, \top)$ forms a commutative monoid.

2. The pair $(\oplus, \rightarrow)$ satisfy an adjoint condition: for all $a, b, c \in L$ we have

$$a \oplus b \leq c \iff a \leq c \rightarrow b.$$

Example 5.1. Any powerset lattice 2^X can be given a residuated structure by defining

$$A \oplus B = A \cap B, \quad A \rightarrow B = \begin{cases} X & A \subseteq B \\ B & \text{otherwise} \end{cases}$$

Example 5.2. Consider the extended non-negative reals $\mathbb{R}_+ = [0, \infty]$ with the reverse ordering, so that $\perp = \infty$, $\top = 0$, and the meet and join are given by the maximum and minimum respectively. The adjoint operators are given by

$$a \oplus b = a + b, \quad a \rightarrow b = \max(0, a - b).$$

An L-**fuzzy set** A in X is a function $A : X \rightarrow L$. We interpret $A(x)$ as the degree to which x is in the fuzzy set A. Note that when $L = 2 = \{0 \leq 1\}$ is the two element Boolean lattice, we recover ordinary set theory.

We call the collection of fuzzy subsets L^X the L-fuzzy power set lattice, which inherits a lattice structure from L. An L-**fuzzy relation** between sets X and Y is an element $R \in L^{X \times Y}$. Similarly to the case of $L = 2$, fuzzy relations induce morphisms between fuzzy power sets.

Proposition 5.1. *Let $R \in L^{X \times Y}$ be an L-fuzzy relation. Then R defines a join preserving map $R : L^X \rightarrow L^Y$ where*

$$R(A)(y) = \bigvee_{x \in X} R(x, y) \oplus A(x)$$

for $A \in L^X$ and $y \in Y$.

The product lattice $L \times K$ has a partially ordering such that $(\ell, k) < (\ell', k')$ when $\ell < \ell'$ and $k < k'$. Lastly, we define the **tensor product** of L-fuzzy power sets to be

$$L^X \otimes L^Y = L^{X \times Y}.$$

Given relations $R \in L^{X \times Y}$ and $T \in L^{Z \times W}$ we define their tensor product to be a relation $R \otimes T \in L^{(X \times Y) \times (Z \times W)}$ by

$$R \otimes T(\langle x, y \rangle, \langle z, w \rangle) = R(x, y) \oplus T(z, w).$$

Sheaf-Theoretic Graph Traversals

Let $G = (V, E)$ be a finite directed graph without self loops. We want to describe traversals over G using sheaves that can be easily generalized to traversals in digraphs with extra structure, for example time-varying graphs. We first make precise what we mean by a traversal in a digraph.

- A **walk** in G is a (possibly infinite) alternating sequence of vertices and edges

$$(v_0, e_0, v_1, e_1, \dots)$$

such that v_i is the source of e_i and the target of e_{i-1}. We may suppress the vertex entries as they are implied by the edges. For fixed vertices v_s and v_t we define the set of based walks $v_s \rightsquigarrow v_t$ by $\text{Walk}(v_s, v_t)$.
- A **path** in G is a walk such that every vertex and edge is used at most once. For fixed vertices v_s and v_t we define the set of based paths $v_s \rightsquigarrow v_t$ by $\text{Path}(v_s, v_t)$.

We now explore lattice structures on graph traversals which we dub **multiwalks** and **multipaths** by viewing them as global sections of certain lattice-valued sheaves.

For a directed graph G, a **cellular sheaf of lattices** consists of the following data:

(1) A lattice $\mathcal{F}(v)$ for each vertex $v \in V$.
(2) A lattice $\mathcal{F}(e)$ for each directed edge $e \in E$.
(3) A join preserving map $\mathcal{F}_{v<e} : \mathcal{F}(v) \rightarrow \mathcal{F}(e)$ for each incidence $v < e$ for $v \in V$ and $e \in E$.

When all stalks are fuzzy power sets $\mathcal{F}(\sigma) = L^{X_\sigma}$, it is enough to specify a relation $\mathcal{F}_{v<e} \in L^{X_v \times X_e}$ for each restriction map. This will be the case for all of the following examples.

We define the 0-**cochains** of $\mathcal{F}$ to be elements of the product lattice

$$C^0(G; \mathcal{F}) := \prod_{v \in V} \mathcal{F}(v).$$

A cochain $x \in C^0(G; \mathcal{F})$ is called a **global section** if for every edge $e = (v, u)$ we have

$$\mathcal{F}_{v<e}(x_v) = \mathcal{F}_{u<e}(x_u).$$

Denote the set of global sections by $\Gamma(G; \mathcal{F})$, which is always nonempty since it contains the bottom cochain $x^\perp$ defined by $(x^\perp)_v = \perp_{\mathcal{F}(v)}$. In [37] it is shown that $\Gamma(G; \mathcal{F})$ is a quasi-sublattice (a subset which is a lattice in its own right) of $C^0(G; \mathcal{F})$. We can lift the tensor product of lattices to lattice sheaves $\mathcal{F}$ and $\mathcal{G}$ over G by defining

$$(\mathcal{F} \otimes \mathcal{G})(\sigma) = \mathcal{F}(\sigma) \otimes \mathcal{G}(\sigma)$$
$$(\mathcal{F} \otimes \mathcal{G})_{v<e} = \mathcal{F}_{v<e} \otimes \mathcal{G}_{v<e}.$$

Later this product will be useful when adding extra traversal conditions or structure to a digraph.

Multipaths

We begin by defining a natural generalization of the path sheaf. Let $G = (V, E)$ be a directed graph with distinguished source vertex v_s and target vertex v_t. For a vertex v, denote the set of **flows** by

$$\text{Flow}(v) = \text{In}(v) \times \text{Out}(v).$$

For $f \in \text{Flow}(v)$ let f_{in} and f_{out} denote the incoming and outgoing edge respectively. The **multipath sheaf** $\mathcal{P}$ has vertex stalks

$$\mathcal{P}(v) = \begin{cases} 2^{\text{In}(v)} & v = v_s \\ 2^{\text{Out}(v)} & v = v_t \\ 2^{\text{Flow}(v)} & \text{otherwise} \end{cases}$$

and edge stalks $\mathcal{P}(e) = 2^{\{e\}} \cong 2$. Define the restriction relations of $\mathcal{P}$ by using the functions that define the path sheaf of [16]. In particular, for $v \in \{v_s, v_t\}$ we define relations

$$\mathcal{P}_{v<e}(x, e) = \begin{cases} 1 & x = e \\ 0 & \text{otherwise} \end{cases}$$

and for all other vertices we define

$$\mathcal{P}_{v<e}(f, e) = \begin{cases} 1 & f_{in} = e \text{ or } f_{out} = e \\ 0 & \text{otherwise} \end{cases}.$$

Example 5.3. Consider the subgraph in Figure 4. At the source vertex v_s and edge e we show the stalk lattices as well as the join preserving map induced by the relation defining $\mathcal{P}_{v_s<e}$.

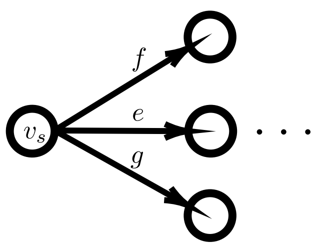


Figure 4: A subgraph with a distinguished source node v_s and three edges $\{f, e, g\}$.

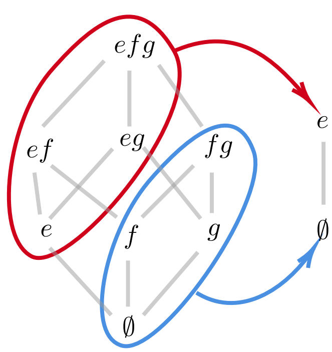


Figure 5: The map $\mathcal{P}_{v_s<e}$ from Example 5.3.

Proposition 5.2. *If G is acyclic, then there is a bijection Atom$(\Gamma(G;\mathcal{P})) \cong$ Path(v_s, v_t).*

See the appendix for a proof of this proposition.

Remark 5.1. An atom of a lattice can be thought of as a generalization of singletons inside a powerset lattice, but note that in general $\Gamma(G;\mathcal{P}) \ncong 2^{\text{Path}(v_s,v_t)}$.

In a graph with cycles, every path from source to target can be realized as a global section, but the converse does not hold. We conclude the discussion of the multipath sheaf by showing it is a geometric lattice. Geometric lattices are noteworthy because they are equivalent to finite simple matroids, which are valuable tools in combinatorial optimization.

Lemma 5.1. *The join of two global sections $x, y \in \Gamma(G;\mathcal{P})$ is given by stalks-wise set union*

$$(x \vee y)_v = x_v \cup y_v.$$

Proof. Note that in $C^0(G;\mathcal{P})$, the join of two cochains is indeed given by the stalk-wise union. The statement follows from the fact $\Gamma(G;\mathcal{P})$ is quasi-sublattice of $C^0(G;\mathcal{P})$ and that the resqutriction maps are join preserving. □

Proposition 5.3. *If G is acyclic, then $\Gamma(G;\mathcal{P})$ is a geometric lattice.*

Proof. It remains to show that $\Gamma(G;\mathcal{P})$ is atomistic and semi-modular. Let $x \neq \emptyset$ be a global section, then for any vertex u and any flow or edge $\phi_0 \in x_u$, we can apply the construction in the previous proposition to build a based path a_ϕ that is a sub-global section of x. If there is a $\phi_1 \in x$ not in this path, then we construct another path a_{ϕ_1} and so on. This process eventually ends, and it follows that $\bigvee_i a_{\phi_i} = x$ by Lemma 5.1.

Given two sections x and y, we show that $y <: x$ if and only if $y < x$ and there is some (not necessarily unique) path $p < x$ such that $y \vee p = x$. Choose a flow or edge $\phi \in x$ not in y and construct an atomic sub-section a_ϕ. By lemma 5.1 $y \vee p \subseteq x$ and hence we must have $y \vee p = x$.

Now suppose $x \wedge y <: x$. Then there is a path $p < x$ with $(x \wedge y) \vee p = x$. It follows that $p \vee y \leq x \vee y$ and if we assume this inequality is strict, then select and edge/flow $\eta \in x \vee y$ that is not in $p \vee y$. Completing η to a sub-section $z < x \vee y$ contradicts the fact $p \vee (x \wedge y) = x$ and hence $p \vee y = x \vee y$ which implies $y <: x \vee y$. □

Weighted Graphs

Let $L = [0, \infty]$ have the residuated structure from Example 5.2 and let G be a *weighted* digraph, that is, G is equipped with a weight function $\omega : E \to L$. We encode the weight structure of the graph using a variant of the multipath sheaf. We will see that these sheaves are useful for computing the lowest cost path from a specified base point. Call a vertex v in G a **sink** if $\text{Out}(v) = \emptyset$ and denote the set of sinks by T. We will assume that G has a distinguished source vertex v_s that is not also a sink. We define the stalks of the **broadcast sheaf** on a cell σ by

$$\mathcal{B}(\sigma) = \begin{cases} L^{\text{Out}(v)} & \sigma = v_s \\ L^{\text{In}(v)} & v \in T \\ L^{\{e\}} & e \in E \\ L^{\text{Flow}(v)} & \text{otherwise} \end{cases}$$

The restriction maps are now given by *fuzzy* relations, so for the source v_s we define the restriction realtion

$$\mathcal{B}_{v_s<e}(x, e) = \begin{cases} \omega(e) & x = e \\ \infty & \text{otherwise} \end{cases}$$

and for the sink nodes $v_t \in T$ we define

$$\mathcal{B}_{v<e}(x,e) = \begin{cases} 0 & x = e \\ \infty & \text{otherwise} \end{cases}.$$

and finally for all other vertices we define

$$\mathcal{B}_{v<e}(f,e) = \begin{cases} 0 & f_{in} = e \\ \omega(e) & f_{out} = e \\ \infty & \text{otherwise} \end{cases}.$$

The fuzzy flow/edge sets over the stalks encode the minimum of the aggregate weights of paths using a particular flow/edge. Even when G is an acyclic graph, the relationship between paths and global sections of $\mathcal{B}$ is more complicated since $\Gamma(G;\mathcal{B})$ is neither finite nor atomic. Despite this, every path in $\mathrm{Path}(v_s, v_t)$ can be realized as a global section.

Proposition 5.4. *Suppose G is an acyclic digraph. Let $x^\top \in \Gamma(G;\mathcal{B})$ be the top element of the global sections lattice and $D : V \to L$ be the function*

$$D(v) = \bigvee_{\phi \in \mathcal{B}(v)} (x^\top)_v(\phi).$$

Then D computes the weight of the shortest path from v_s to v.

Proof. Fix a vertex $v = v_n$ and let $p = (v_0, e_0, v_1 \dots, e_{n-1}, v_n)$ be the shortest cost path $v_s \rightsquigarrow v$. We construct a cochain x^p as follows. If v is a transshipment node, choose any $e_n \in \mathrm{Out}(v)$ and let $x^p_{v_s}(e_0) = \mathbb{1}_{e_0}$ and $x^p_{v_k}(e_{k-1}, e_k) = \sum_{i=0}^{k} \omega(e_i)$ for $k = 1, \dots n$. Extend the cochain using e_n and any flow (e_n, e_{n+1}) and continue this process until a sink is reached. Then the resulting cochain will be global section and hence $x^p \leq x^\top$. A similar argument holds when v is a sink or the source.

We claim that $D(v_n) = x_{v_n}(e_n, e')$. Indeed suppose there is a section y with $y_{v_n} > x_{v_n}$ with a flow $y_{v_n}(f) > x_{v_n}(e_n, e')$. Tracing back through y_u to the source using this flow will produce a shorter cost path than p yielding a contradiction. ☐

Multiwalks

In this section, we define a sheaf whose atomic global sections correspond to based walks from distinguished source and target vertices. We associate to each cell $\sigma \in V \cup E$

$$X_\sigma = \begin{cases} \big[\mathrm{Out}(v_s) \amalg \mathrm{Flow}(v_s)\big] \times \mathbb{N} & \sigma = v_s \\ \big[\mathrm{In}(v_t) \amalg \mathrm{Flow}(v_t)\big] \times \mathbb{N} & \sigma = v_t \\ \mathrm{Flow}(\sigma) \times \mathbb{N} & \sigma \in V \setminus \{v_s, v_t\} \\ \mathbb{N} & \sigma \in E \end{cases}$$

and define the stalks of $\mathcal{W}$ to be

$$\mathcal{W}(\sigma) = 2^{X_\sigma}.$$

We define the restriction relations in three cases. For the source vertex v_s we let

$$\mathcal{W}_{v_s \leq e}(\langle \phi, n \rangle, m) = \begin{cases} 1 & \phi = e,\ n = m-1 = 0 \\ 1 & \phi_s = e,\ n = m-1 \\ 1 & \phi_t = e,\ n = m \\ 0 & \text{otherwise} \end{cases}$$

for the target vertex we let

$$\mathcal{W}_{v_t \leq e}(\langle \phi, n \rangle, m) = \begin{cases} 1 & \phi = e,\ n = m \\ 1 & \phi_s = e,\ n = m-1 \\ 1 & \phi_t = e,\ n = m \\ 0 & \text{otherwise} \end{cases}$$

and finally for a transshipment node we let

$$\mathcal{W}_{v \leq e}(\langle \phi, n \rangle, m) = \begin{cases} 1 & \phi_s = e,\ n = m-1 \\ 1 & \phi_t = e,\ n = m \\ 0 & \text{otherwise} \end{cases}$$

where ϕ is either an edge or flow and n and m are natural numbers. As we will see, atomic sections of this sheaf will correspond to walks in G, with the caveat that our definition of walk includes infinite walks. For a cochain $x \in C^0\mathcal{W}$ we define its cardinality by

$$|x| = \sum_{v \in V} |x_v|.$$

Denote the sublattice of finite chains $|x| < \infty$ by $C^0_{\mathsf{fin}}(G;\mathcal{W}) \subseteq C^0(G;\mathcal{W})$ and finite sections by $\Gamma_{\mathsf{fin}}(G;\mathcal{W}) \subseteq \Gamma(G;\mathcal{W})$.

Proposition 5.5. *There is a bijection $\mathit{Atom}(\Gamma_{\mathit{fin}}(G;\mathcal{W})) \cong \mathit{Walk}(v_s, v_t)$.*

Proof. Let $w = (e_0, e_1, \dots, e_n)$ be walk from v_s to v_t. We construct a cochain x^w whose stalk data at v is the union of every flow $(\langle e_i, e_{i+1} \rangle, i+1)$ through v. Additionally for the source and target vertex we add to their respective unions the elements $(e_0, 0)$ and (e_n, n). It is easy to check x^w is both finite and a global section and moreover that this mapping is injective. Surjectivity is proven similarly to Proposition 5.2 once one shows that for every atom a and $k \in \mathbb{N}$ there is at most one edge/flow ϕ with $(\phi, k) \in a$.

To see x^w is atomic, let $y < x^w$ be a sub-section of x^w. Without loss of generality assume there is a transshipment node u such that $y_u \subset x^w_u$. Then there is a flow $(\langle e_i, e_{i+1} \rangle, i+1)$ not present in y, but it follows that $\mathcal{W}_{v \leq e_i}(y_u) \neq \mathcal{W}_{u \leq e_i}(y_v)$ for $e_i = (v, u)$ unless $(\langle e_{i-1}, e_i \rangle, i) \notin y_v$. By induction one arrives at $y = \emptyset$ and hence x^w is atomic. ☐

Temporal Networks

A **time-varying graph** (TVG) is a tuple $\mathfrak{G} = (V, E, T, \rho, \zeta)$ where $G = (V, E)$ is the underlying directed graph, T is the time domain (typically $\mathbb{N}$ or $\mathbb{R}_{\geq 0}$), together with the following functions.

- The **presence function** $\rho : E \times T \to 2$ of available times for each edge.
- The **latency function** $\zeta : E \times T \to T$ of traversal times for each edge.

A **journey** in a TVG $\mathfrak{G}$ is a sequence

$$J = \langle (e_0, t_0), (e_1, t_1), \dots, (e_n, t_n) \rangle$$

such that $(e_0, \dots, e_n)$ is a walk in the underlying digraph G, $\rho(e_i, t_i) = 1$ and $t_{i+1} \geq t_i + \zeta(e_i, t_i)$ for all i.

We encode the structure of a TVG into a sheaf over the underlying directed graph as follows. Let

$$H_T = \{(t, t') \in T^2 \mid t < t'\}.$$

The **time sheaf** $\mathcal{T}$ on the underlying digraph G has stalks

$$\mathcal{T}(\sigma) = \begin{cases} T & \sigma \in \{v_s, v_t\} \cup E \\ H_T & \text{otherwise} \end{cases}$$

which can be interpreted as receipt or delivery times of a signal being sent over an edge. The relations defining the restriction maps are given in the following cases. For the source vertex let

$$\mathcal{T}_{v_s<e}(t, t') = \begin{cases} 1 & \rho(e,t) = 1,\ \zeta(e,t) = t' \\ 0 & \text{otherwise} \end{cases},$$

for the target vertex let

$$\mathcal{T}_{v_t<e}(t, t') = \begin{cases} 1 & \rho(e,t) = 1,\ t = t' \\ 0 & \text{otherwise} \end{cases}$$

and finally for the transshipment nodes let

$$\mathcal{T}_{v<e}(\langle t_1, t_2\rangle, t') = \begin{cases} 1 & e = (v,u),\ \rho(e,t_2) = 1,\ \zeta(e,t_2) = t' \\ 1 & e = (u,v),\ \rho(e,t_1) = 1,\ t_1 = t' \\ 0 & \text{otherwise} \end{cases}$$

The **multijourney sheaf** is defined to be the tensor product $\mathcal{J} = \mathcal{T} \otimes \mathcal{W}$.

Proposition 5.6. *Let $\mathfrak{G}$ be a TVG, $\mathcal{T}$ be its corresponding time sheaf, and $\mathcal{W}^{v_s \rightsquigarrow v_t} = \mathcal{W}$ a mulitwalk sheaf on the underlying digraph G. Then there is a bijection $\text{Atom}(\Gamma_{\mathit{fin}}(G; \mathcal{T} \otimes \mathcal{W})) \cong \text{Journey}(v_s, v_t)$.*

Proof. When $L = 2$, the tensor product of two sheaves essentially implements the logical 'and' for pairs of data. That is to say for

$$\begin{aligned} \mathcal{J}_{v<e}(A) &= \{(e,t) \in \{e\} \times T \mid \exists(\phi,\tau) \in A, \\ &\qquad \mathcal{J}_{v<e}(\langle \phi, \tau\rangle, \langle e, t\rangle) = 1\} \\ &= \{(e,t) \in \{e\} \times T \mid \exists(\phi,\tau) \in A, \\ &\qquad \mathcal{T}_{v<e}(\tau, t) = 1 \text{ and } \mathcal{W}_{v<e}(\phi, e) = 1\}. \end{aligned}$$

Hence a cochain $x \in C^0(G; \mathcal{J})$ is a global section if and only if both of its projections into $C^0(G; \mathcal{T})$ and $C^0(G; \mathcal{W})$ are global sections. It follows that the same is true for atoms. One can check that the projection of x into $\Gamma(G; \mathcal{W})$ determines the edges and indices in the definition of a journey. Similarly, the projection into $\Gamma(G; \mathcal{T})$ determines the corresponding times between each edge. The remainder of the proof is similar to Proposition 5.5. □

Laplacian Dynamics

Similarly to vector space valued sheaves, lattice-valued sheaves have their own Laplacian operator. To define it, we need a notion of a *transpose* for a join preserving map.

Proposition 5.7. *Every join preserving map $R : L \to K$ determines a unique meet preserving map $R^+ : K \to L$ such that*

$$R(\ell) \leq k \iff \ell \leq R^+(k)$$

for al $\ell \in L$ and $k \in K$.

When one views the lattices L and K as categories, the above proposition is a special case of Freyd's adjoint functor theorem. The pair (R, R^+) is known as a (monotone) *Galois connection* between L and K.

Definition 5.3. The **Tarski Laplacian** of a lattice-valued sheaf $\mathcal{F}$ is an operator $L : C^0(G; \mathcal{F}) \to C^0(G; \mathcal{F})$, which is defined on a cochain $x \in C^0(G; \mathcal{F})$ at the stalk $\mathcal{F}(v)$ by

$$(Lx)_v = \bigwedge_{e \in \delta v} \mathcal{F}^+_{v<e} \left(\bigwedge_{w \in \partial e} \mathcal{F}_{w<e}(x_w) \right).$$

Sheaf Laplacians are especially useful for computing global sections. Given an arbitrary cochain $x \in C^0(G; \mathcal{F})$ of a lattice sheaf $\mathcal{F}$, there is an associated discrete-time dynamical system we call the **harmonic flow**

$$x_{t+1} = x_t \wedge Lx_t$$

with initial condition $x_0 = x$. It is shown in [37] that, assuming some finiteness conditions, the above system converges to a global section $x^* \in \Gamma(G; \mathcal{F})$. Moreover, sheaf Laplacian dynamics update node states using information locally available to the node and hence each evaluation of Lx_t can be computed with a distributed algorithm. We suspect the Tarski Laplacian could aid computing global sections given a contact schedule for a time-varying graph and leave such examples for a future work.

6. GLUING OF LATTICE SHEAVES

In multidomain routing, a global network may be patched together out of subnetworks. Cellular sheaves facilitate the gluing of local data on these subnetworks into global data on the global network. In [2] a process was described for gluing together set-valued path sheaves on so-called *sequential networks*. In that paper, the authors leveraged the pullback operation to glue the path sheaves on the subnetworks together. In this section, we describe an example of a similar process for lattice path sheaves.

Consider the network shown in Figure 6. This network consists of two overlappig subnetworks G_1 and G_2. The subnetworks overlap on a two disjoint nodes.

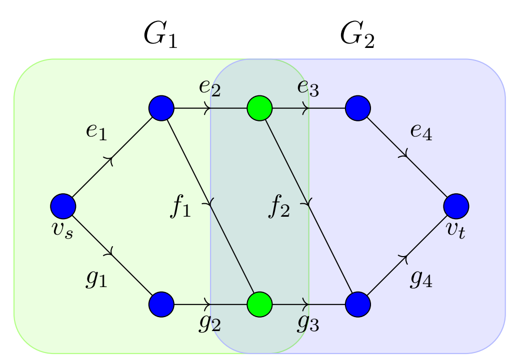


Figure 6

We define cellular sheaves of lattices $\mathcal{P}_i$ on G_i, for $i = 1, 2$,

as follows. Then we set the vertex stalks of $\mathcal{P}_1$ to be

$$\mathcal{P}_1(v) = \begin{cases} 2^{\text{Out}(v_s)}, & v = v_s \\ 2^{\text{In}(v)}, & v \in G_1 \cap G_2 \\ 2^{\text{Flow}(v)}, & \text{otherwise} \end{cases}$$

and the vertex stalks of $\mathcal{P}_2$ to be

$$\mathcal{P}_1(v) = \begin{cases} 2^{\text{Out}(v)}, & v \in G_1 \cap G_2 \\ 2^{\text{In}(v_t)}, & v = v_t \\ 2^{\text{Flow}(v)}, & \text{otherwise} \end{cases}.$$

For $i = 1, 2$, we let the edge stalks of $\mathcal{P}_i$ be $2^{\{e\}}$ for each edge e in G_i. The restriction maps are defined exactly how they were for the multipath sheaf, which is defined in Section 5. Furthermore, we can extend $\mathcal{P}_1$ over G_2 and $\mathcal{P}_2$ over G_2 trivially; we define the stalks of $\mathcal{P}_i$ to be the trivial lattice over each vertex and edge of $G_1 \cup G_2$ that does not lie in G_i.

Additionally, we define the **overlap sheaf** $\mathcal{O}$ on $G_1 \cup G_2$. Over each vertex in $G_1 \cap G_2$, we define the stalk of $\mathcal{O}$ to be $2^{\{\top\}}$. The stalks of $\mathcal{O}$ over all other vertices and over all edges of $G_1 \cup G_2$ are taken to be the trivial lattice.

To define the pullback $\mathcal{P}_1 \times_{\mathcal{O}} \mathcal{P}_2$ we need maps $\phi_i : \mathcal{P}_i \to \mathcal{O}$. The maps ϕ_i, $i = 1, 2$, are determined by their values on the stalks over the overlap $G_1 \cap G_2$, which are

$$\phi_i(\{e\}) = \{\top\}, \qquad \phi_i(\emptyset) = \emptyset$$

for any edge e either in $\text{In}(v)$ or $\text{Out}(v)$, for $v \in G_1 \cap G_2$. We define ϕ_i to be join preserving, so that $\phi_i(e \vee f) = \phi_i(e) \vee \phi_i(f)$ for any edges e and f.

With this data in place, we can compute the pullback. If L_1 and L_2 are two lattices, we define their product $L_1 \times L_2$ as the usual Cartesian product of L_1 and L_2 with lexicographic ordering. Then the pullback of the $\mathcal{P}_i$ over $\mathcal{O}$ is given by

$$\mathcal{P}_1 \times_{\mathcal{O}} \mathcal{P}_2 = \{(a, b) \in \mathcal{P}_1 \times \mathcal{P}_2 | \phi_1(a) = \phi_2(b)\}.$$

It turns out that $\mathcal{P}_1 \times_{\mathcal{O}} \mathcal{P}_2$ is isomorphic to the multipath sheaf $\mathcal{P}$ on $G_1 \cup G_2$, with source v_s and target v_t.

For example, let x denote the upper green vertex in $G_1 \cap G_2$, as in Figure 6. Then the stalks $\mathcal{P}_i(x)$ are given by

$$\mathcal{P}_1(x) = 2^{\{e_2\}}, \qquad \mathcal{P}_2(x) = 2^{\{e_3, f_2\}}.$$

For the overlap sheaf, the stalk is

$$\mathcal{O}(x) = 2^{\{\top\}}.$$

Then the stalk of the pullback sheaf over x is the lattice

$$\{(\{e_2\}, \{e_3, f_2\}), (\{e_2\}, \{e_3\}), (\{e_2\}, \{f_2\}), (\emptyset, \emptyset)\}.$$

This lattice is isomorphic to the lattice $2^{\text{Flow}(x)}$, which is precisely the stalk of the multipath sheaf over x.

This example illustrates how we can utilize the pullback operation to glue together sheaves of lattices on overlapping networks to obtain a sheaf of lattices on the union of the networks. It is our hope that we can bring extra structure into this construction. For example, we are interested in computing the pullback of *residuated lattices*. More research is required on this front.

Remark 6.1. In this section, we gave an example of the pullback of sheaves of lattices. The pullback construction can be applied for sheaves taking values in other categories. In [2] the authors describe pullbacks of set-valued path sheaves on so-called *sequential networks*. The pullback construction may also be applied to vector space path sheaves.

7. Degenerate Global Sections

In [2] it is shown that the free vectorization of the set-valued path sheaf produces global sections that cannot be interpreted as paths in the underlying graph. More generally, it may be the case that extra constraints are necessary that cannot be fully encoded in a single cellular sheaf.

We first review **cellular cosheaves**, the dual notion of a cellular sheaf. Cosheaves in their own right have seen fascinating applications in graph statics [38] and topological data analysis [39]. Over a graph G, a cellular cosheaf $\mathcal{G}$ of vector spaces consists of:

(1) A vector space $\mathcal{G}(v)$ for each vertex $v \in V$.
(2) A vector space $\mathcal{G}(e)$ for each edge $e \in E$.
(3) A linear map

$$\mathcal{G}_{v<e} : \mathcal{G}(e) \to \mathcal{G}(v)$$

whenever we have the incidence relation $v < e$ for $v \in V$ and $e \in E$.

Note that the only difference from a cellular sheaf is that the linear maps between stalks go in the other direction.

Ends and Coends

In the previous sections, cellular sheaves were defined as assignments of cells $\sigma \in G$ to objects like sets, vector spaces, or lattices along with assignments of incidences $v < e$ to transformations between these objects. This is exactly the data of a *covariant functor* from the poset of cells in G to the category of sets, vector spaces or lattices. Similarly a cosheaf is exactly the data of a *contravariant functor*. The global (co)sections of these functors are given by the *limit* $\lim \mathcal{F} = \Gamma(G; \mathcal{F})$ and *colimit* $\text{colim}\, \mathcal{G} = \Gamma(G; \mathcal{G})$. Later we will see that we can assemble a sheaf and cosheaf together into what is known as a *bifunctor*.

For the remainder of this section we will want our (co)sheaves to be valued in a **monoidal category** $(C, \otimes, I)$, i.e. a category equipped with a binary operation $\otimes$ and a unit object such that $A \otimes I \cong I \otimes A \cong A$ for every object $A \in C$. One also requires various coherences one can find in [40]. A **bifunctor** over G valued in $(C, \otimes, I)$ consists of:

1. Objects $\mathcal{B}(\sigma, \tau) \in C$ for each pair of cells $(\sigma, \tau) \in (V \cup E)^2$.
2. Restriction morphisms $\alpha_{\sigma < \sigma'} : \mathcal{B}(\sigma, \tau) \to \mathcal{B}(\sigma', \tau)$ for each incidence $\sigma < \sigma'$.
3. Extension morphisms $\alpha_{\tau' < \tau} : \mathcal{B}(\sigma, \tau) \to \mathcal{B}(\sigma, \tau')$ for each incidence $\tau' < \tau$.

In other words, $\mathcal{B}$ acts like a cellular sheaf in the first component and a cellular cosheaf in the second. The **end** of a bifunctor $\mathcal{B}$ over G is the equalizer

$$\int_G \mathcal{B}(\sigma, \sigma) = \text{eq} \left(\prod_{\sigma \in V \cup E} \mathcal{B}(\sigma, \sigma) \rightrightarrows \prod_{\sigma \leq \sigma'} \mathcal{B}(\sigma', \sigma) \right)$$

where the two arrows in the diagram represent the two possible mappings

$$\mathcal{B}(\sigma,\sigma) \to \mathcal{B}(\sigma',\sigma) \leftarrow \mathcal{B}(\sigma',\sigma').$$

The dual notion of the **coend** is defined by

$$\int^{G} \mathcal{B}(\sigma,\sigma) = \operatorname{coeq}\left(\prod_{\sigma\in V\cup E} \mathcal{B}(\sigma,\sigma) \leftleftarrows \prod_{\sigma\leq\sigma'} \mathcal{B}(\sigma,\sigma') \right)$$

where the two arrows in the diagram represent the two possible mappings

$$\mathcal{B}(\sigma',\sigma) \to \mathcal{B}(\sigma,\sigma) \leftarrow \mathcal{B}(\sigma,\sigma').$$

For more details on (co)ends see [41].

For a sheaf-cosheaf pair $\mathcal{F}$ and $\mathcal{G}$ over G, we assemble them into a bifunctor as follows. Suppose we are further given two families of maps α and β with

$$\alpha_{\sigma\leq\sigma',\tau} : \mathcal{F}(\sigma)\otimes\mathcal{G}(\tau) \to \mathcal{F}(\sigma')\otimes\mathcal{G}(\tau)$$
$$\beta_{\sigma,\tau'\leq\tau} : \mathcal{F}(\sigma)\otimes\mathcal{G}(\tau) \to \mathcal{F}(\sigma)\otimes\mathcal{G}(\tau')$$

then the collection $(\mathcal{F},\mathcal{G},\alpha,\beta)$ determines a bifunctor $\mathcal{B}$ with

$$\mathcal{B}(\sigma,\tau) = \mathcal{F}(\sigma)\otimes\mathcal{G}(\sigma)$$

and restriction/extensions morphisms given by α and β.

Example 7.1. In any monoidal category C, we can obtain a bifunctor $(\mathcal{F}\otimes\mathcal{G})(\sigma,\tau) = \mathcal{F}(\sigma)\otimes\mathcal{G}(\tau)$ using the restriction and extension maps given by

$$\alpha_{\sigma<\sigma',\tau} = \mathcal{F}_{\sigma<\sigma'}\otimes \mathrm{id}$$
$$\beta_{\sigma,\tau'<\tau} = \mathrm{id}\otimes\mathcal{G}_{\tau'<\tau}.$$

When C is cartesian monoidal, it follows that

$$\int_G \mathcal{F}\otimes\mathcal{G} \cong \lim\mathcal{F}\otimes\lim\mathcal{G}$$

since limits commute with one another. Now let C be the category of real vector spaces with tensor product as the monoidal product. Let $\mathcal{S}_\sigma$ denote the skyscraper cosheaf at σ, that is

$$\mathcal{S}_\sigma(\tau) = \begin{cases}\mathbb{R} & \sigma=\tau\\ 0 & \text{otherwise}\end{cases}$$

with all extension maps being 0. Then one can show

$$\int_G \mathcal{F}\otimes\mathcal{S}_\sigma \cong \mathcal{F}(\sigma).$$

For more details on this particular example see [42].

Example 7.2. Every sheaf $\mathcal{F}$ can be construed as bifunctor that is 'mute' in the second argument. That is for every pair (σ,τ) we can define a bifunctor $\hat{\mathcal{F}}(\sigma,\tau) = \mathcal{F}(\sigma)\otimes\mathcal{G}(\tau)$ where $\mathcal{G}$ is a **constant sheaf**, defined by $\mathcal{G}(\sigma) = I$ for all σ and all extensions maps being the identity. One can show that the global sections of $\mathcal{F}$ can be expressed by the end:

$$\Gamma(G;\mathcal{F}) \cong \int_G \hat{\mathcal{F}}(\sigma,\sigma).$$

This motivates our intuition that ends of the bisheaves constructed above can be thought of as global sections of $\mathcal{F}$ with additional constraints given encoded in $\mathcal{G}$, α and β.

Example 7.3. Let C be the category of real vector spaces with the monoidal product given by the direct sum. For any sheaf $\mathcal{F}$ on G, let $\mathcal{G}$ be defined by

$$\mathcal{G}(\sigma) = \begin{cases}\mathbb{R} & \sigma\in V\\ 0 & \sigma\in E\end{cases}$$

so that all extension maps must be zero. We define the relevant α and β families by the matrices

$$\alpha_{v\leq e,\tau} = \begin{bmatrix}\mathcal{F}_{v<e} & 0\\ \vec{1} & 0\end{bmatrix} \qquad \beta_{v\leq e,\tau} = \begin{bmatrix}\mathrm{id} & 0\\ 0 & 0\end{bmatrix}$$

such that $\vec{1}$ is vector of ones and id is the identity matrix of the appropriate dimensions. Intuitively, the end $\int_G \mathcal{F}\otimes\mathcal{G}$ computes those global sections $x\in\Gamma(G;\mathcal{F})$ such that $x_v\cdot\vec{1} = 0$.

8. Algebro-geometric Modeling

Modeling temporal networks using algebro-geometric methods seems to be particularly appealing as this would allow for the full breadth of algebraic geometry theory to be brought to bear against problems in delay-tolerant networking. Indeed, a hint was given by Definition 2.2 which suggests the usage of *moduli spaces* of metric spaces.

A first step in this direction is taken in [14], where the possibility of using graph varieties to model temporal networks is explored. We briefly recall the main construction here. Fix an algebraically closed field $\Bbbk$ and let $G = (V,E)$ be an undirected graph. Define

$$\mathrm{Gr}(G) = \prod_{w\in V}\mathbb{P}^2 \times \prod_{e\in E}\widehat{\mathbb{P}^2}$$

where $\widehat{\mathbb{P}^2}$ is the dual of $\mathbb{P}^2$ over $\Bbbk$. Informally, a point in $\mathrm{Gr}(G)$ is simply the data of a point in $\mathbb{P}^2$ for each vertex of G together with a line in $\mathbb{P}^2$ for each edge of G. The space $\mathrm{Gr}(G)$ is naturally parametrized by $V\times E$. Given a point $P\in\mathrm{Gr}(G)$, if for all $v\in V$ and $e\in E$, $v\in e$ implies that $P(v)\in P(e)$, then P is called a *picture* of G. We denote by $\chi(G)$ the collection of all such pictures of G. As $\chi(G)$ can be defined using Plücker coordinates, it is Zariski-closed in $\mathrm{Gr}(G)$ and is thus a quasiprojective variety. Morally, $\chi(G)$ parametrizes the various embeddings of G in $\mathbb{P}^2$ in a way that preserves the graph structure of G; thus it is a natural algebro-geometric realization of G. In [43], properties of the graph G are translated into properties of the variety $\chi(G)$ and vice-versa.

The motivation behind exploring this construction in [14] is to translate connectivity properties of the graph G into properties of the structure sheaf on $\chi(G)$, with the goal of mimicking the pathfinding sheaf defined in [16] in an algebro-geometric context. However, the Zariski topology is too coarse for the structure sheaf on $\chi(G)$ to encode any useful topological information about G. The most natural solution to this obstruction is to use a different Grothendieck topology on the category of $\Bbbk$-schemes. We pose here some potential research questions that we hope to explore in future works.

Question 8.1. A $\mathcal{C}$-valued sheaf on the discrete site over the category of $\Bbbk$-schemes associates to each picture $P \in \chi(G)$ an object in $\mathcal{C}$. Is there a coarser Grothendieck topology that we can impose on the category of $\Bbbk$-schemes that still allows such an association?

Question 8.2. Viewing a temporal network G_t as a one-parameter family of graphs parametrized by t, can we glue the picture varieties $\chi(G_t)$ together along some open subsets of the $\chi(G_t)$ in such a way that the structure sheaf on this glued scheme gives insight into connectivity properties of the temporal network G_t?

One possible alternative way to model temporal graphs using algebro-geometric tools is via moduli spaces. More specifically, one could reasonably view a temporal graph as a path in a moduli space of graphs. This leads naturally to the following question.

Question 8.3. How should an appropriate moduli space of graphs be constructed? What is its scheme structure? What is its stack structure (if any)?

9. CONCLUSION

The advent of different ways to detect temporal structure in a temporal network gives us new ways to define and classify substructures. This means that simulations or even historical network connectivity data can be reinterpreted to show

- what structures arise naturally,
- which portions of the network are well-connected, and
- which portions of the network might need bolstering.

No matter which method is used to detect domains, these domains can have routing algorithms expressed over them as sheaves, which can then be united into a bigger network. Suggestions for future work are given below.

Future Works

Temporal Curvature and Persistent Homology—Can we detect temporal curvature using persistence? It is known that persistence detects the curvature of discs embedded in different manifolds [44], but these results are for a very different setting than ours. Is there a sense in which zigzag persistence detects temporal curvature?

Optimization— Submodular functions are an important class of functions in combinatorial optimization and machine learning, which are defined on power set lattices $f : L^X \to \mathbb{R}$. Given a sheaf of powerset lattices $\mathcal{F}$ and a collection of submodular functions $f_v : L^{X_v} \to \mathbb{R}$ for each vertex, one wishes to solve

$$\min_{x \in \Gamma(G;\mathcal{F})} \sum_{v \in V} f_v(x_v).$$

Recent work in submodular minimization [45] suggests there may be probabilistic and distributed algorithms for the above problem.

Path Finding in a TVG—Let $\mathfrak{G}$ be a TVG with distinguished source and target nodes v_s and v_t. We would like to explore using the harmonic flow of the multijourney sheaf $\mathcal{J}$ to compute all journeys from $v_s \rightsquigarrow v_t$ given a schedule of contacts between nodes. However, to ensure convergence we will need to augment our definition of $\mathcal{T}$ and $\mathcal{W}$ to adhere to finiteness conditions of [37]. One can accomplish this by using finite lattices for every stalk, or in other words have $\mathcal{J}(\sigma) = L^{X_\sigma}$ for $|X_\sigma| < \infty$. This means we will need to replace our set of times T by, say, a finite cover of $[0, \infty]$ for the time sheaf $\mathcal{T}$ and replace $\mathbb{N}$ by a finite set $[N] = \{0, 1, \ldots N\}$, as well as some slight alterations to the relations defining the restriction maps.

Network Coding and Data Flow Optimization— Network coding made its debut in 2000 by Ahlswede et al. in [46] as a form of a way to achieve the information-theoretic analog of the max-flow-min-cut theorem that they prove in their paper for multicast (single-source multi-target). Network coding is known to increase throughput in data communication, and linear coding schemes are sufficient for achieving the most optimal solution [47]. This motivates the study of network coding for data flow optimization on DTN (mostly as random linear network coding) [48–50], as well as satellite communication (mostly as physical-layer network coding) [51–53].

However, network coding appears to provide benefit in throughput, only in the case of single multicast [54, 55]. That is, it is not clear whether or not network coding would be more useful than routing for every method of data delivery. A long-standing open problem in information theory known as the *multiple unicast conjecture* (or *the undirected k-pairs conjecture*) is stated as follows:

Imported Conjecture 1 (Li, Li '06 [54–56])**.** *The information flow rate achieved by network coding is equal to the fractional multicommodity-flow rate in undirected multiple unicast networks.*

In other words, network coding may not have any benefit over packet routing; more specifically, undirecting the edges of the network is as good as doing network coding [56]. There have been many related works and progress on this conjecture over the past years. Harvey et al. [57, 58] provides an example class of graphs where the conjecture is true. Braverman, Garg, and Schvartzman [59] show by construction that if the conjecture is false, then the advantage of network coding can be amplified poly-logarithmically. Reversely, Afshan et al. [60] proved that if the conjecture turns out to be true, the (circuit size) lower bound of multiplication can be improved up to $\Omega(n \log n)$.

It turns out network coding can also be modeled using the language of sheaves. Ghrist and Hiraoka [61, 62] defined multicast network coding sheaf as a cellular sheaf[2] that equips every edge e with a vector space of dimension $\mathrm{cap}(e)$ (its capacity), vertex v with dimension the sum of all incoming $\mathrm{cap}(e)$'s, and define the restriction map as the canonical projection map (so the map going out from a vertex v to an outgoing edge e represents the encoding-then-forwarding process within v).

Imported Theorem 1 (Ghrist, Hiraoka [61, 62] (Informal))**.** *The maximum dimension of the* 0*-th cohomology* H^0 *of multicast network coding sheaf is equal to the maximum information flow of the network.*

Given the conjecture, an interesting generalization from this theorem would be to extend this result to the 'multiple-unicast' network coding sheaf. One could start by reducing

[2] The concept of cellular sheaf was not well-defined/known back when Ghrist and Hiraoka wrote their papers. See [63, Chapter 9] and [64, Chapter 9] instead for the definition in terms of the cellular sheaf.

the multiple unicast problem into the multicast problem by adding an imaginary node that is connected with the senders. Moreover, the works of Chambers, Erickson, and Nayyeri [65] and Ghrist and Krishnan [66, 67] provide analogs of max-flow-min-cut problems from (co)homological and sheaf-theoretic perspectives. Extending their results to multicommodity flows and comparing the (dimensions of) cohomologies with the 'multiple-unicast' sheaf cohomologies could lead to a new, topological way to approach the conjecture.

Another viable direction to this conjecture would be to utilize the network embedding technique that was developed in this paper. Upon isoperimetric embedding of a network (graph) on a (Riemannian) manifold, can the process of network coding be modeled as a diffusion process (or some other problem)? Similarly, can the shortest-path problem be reduced to the problem of computing the solution to the Euler-Lagrange equation? This direction of research was first formally studied by Fang in [68]; in particular, [68] proves that proving the conjecture on Riemannian manifold (after embedding) is sufficient.

References

[1] D. J. Israel, K. D. Mauldin, C. J. Roberts, J. W. Mitchell, A. A. Pulkkinen, L. V. D. Cooper, M. A. Johnson, S. D. Christe, and C. J. Gramling, "Lunanet: a flexible and extensible lunar exploration communications and navigation infrastructure," in *2020 IEEE Aerospace Conference*, 2020, pp. 1–14.

[2] A. Hylton, N. Tsuei, M. Ronnenberg, J. Hwang, B. Mallery, J. Quartin, C. Levaunt, and J. Quail, "Advances in modeling solar system internet structures and their data flows," in *2023 IEEE Aerospace Conference*, 2023, pp. 1–20.

[3] Y. Ollivier, "Ricci curvature of markov chains on metric spaces," *Journal of Functional Analysis*, vol. 256, no. 3, pp. 810–864, 2009.

[4] J. Sia, E. Jonckheere, and P. Bogdan, "Ollivier-ricci curvature-based method to community detection in complex networks," *Scientific reports*, vol. 9, no. 1, p. 9800, 2019.

[5] P. van der Hoorn, W. J. Cunningham, G. Lippner, C. Trugenberger, and D. Krioukov, "Ollivier-ricci curvature convergence in random geometric graphs," *Physical Review Research*, vol. 3, no. 1, p. 013211, 2021.

[6] F. Bauer, J. Jost, and S. Liu, "Ollivier-ricci curvature and the spectrum of the normalized graph laplace operator," *arXiv preprint arXiv:1105.3803*, 2011.

[7] F. Mémoli, Z. Wan, and Y. Wang, "Persistent Laplacians: Properties, algorithms and implications," *SIAM Journal on Mathematics of Data Science*, vol. 4, no. 2, pp. 858–884, 2022. [Online]. Available: https://doi.org/10.1137/21M1435471

[8] C. Villani *et al.*, *Optimal transport: old and new*. Springer, 2009, vol. 338.

[9] L. Lu and Z. Wang, "Concentration inequalities in spaces of random configurations with positive ricci curvatures," *arXiv preprint arXiv:1906.03550*, 2019.

[10] R. Bubley and M. Dyer, "Path coupling: A technique for proving rapid mixing in markov chains," in *Proceedings 38th Annual Symposium on Foundations of Computer Science*. IEEE, 1997, pp. 223–231.

[11] C.-C. Ni, Y.-Y. Lin, F. Luo, and J. Gao, "Community detection on networks with ricci flow," *Scientific reports*, vol. 9, no. 1, p. 9984, 2019.

[12] M. E. Newman, "Spectral methods for community detection and graph partitioning," *Physical Review E*, vol. 88, no. 4, p. 042822, 2013.

[13] F. Liu, D. Choi, L. Xie, and K. Roeder, "Global spectral clustering in dynamic networks," *Proceedings of the National Academy of Sciences*, vol. 115, no. 5, pp. 927–932, 2018.

[14] A. Hylton, R. Short, J. Cleveland, O. Freides, Z. Memon, R. Cardona, R. Green, J. Curry, S. Gopalakrishnan, D. V. Dabke, B. Story, M. Moy, and B. Mallery, "A survey of mathematical structures for lunar networks," in *2022 IEEE Aerospace Conference (AERO)*, 2022, pp. 1–17.

[15] G. Carlsson and V. de Silva, "Zigzag persistence," *Foundations of Computational Mathematics*, vol. 10, no. 4, pp. 367–405, 2010.

[16] M. Moy, R. Cardona, R. Green, J. Cleveland, A. Hylton, and R. Short, "Path optimization sheaves," 2020.

[17] R. Short, A. Hylton, J. Cleveland, M. Moy, R. Cardona, R. Green, J. Curry, B. Mallery, G. Bainbridge, and Z. Memon, "Sheaf theoretic models for routing in delay tolerant networks," in *2022 IEEE Aerospace Conference (AERO)*, 2022, pp. 1–19.

[18] R. Short, A. Hylton, R. Cardona, R. Green, G. Bainbridge, M. Moy, and J. Cleveland, "Towards sheaf theoretic analyses for delay tolerant networking," in *2021 IEEE Aerospace Conference (50100)*, 2021, pp. 1–9.

[19] J. Hansen and R. Ghrist, "Toward a spectral theory of cellular sheaves," *Journal of Applied and Computational Topology*, vol. 3, no. 4, pp. 315–358, aug 2019. [Online]. Available: https://doi.org/10.1007%2Fs41468-019-00038-7

[20] X. Wei and G.-W. Wei, "Persistent sheaf Laplacians," 2022.

[21] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Galligher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, Sep. 2008. [Online]. Available: https://ojs.aaai.org/index.php/aimagazine/article/view/2157

[22] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," vol. 58, no. 7, p. 1019–1031, may 2007.

[23] M. E. Aktas, E. Akbas, and A. E. Fatmaoui, "Persistence homology of networks: methods and applications," *Applied Network Science*, vol. 4, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:198147820

[24] V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena, "Statistically significant detection of linguistic change," in *Proceedings of the 24th International Conference on World Wide Web*, ser. WWW '15. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2015, p. 625–635. [Online]. Available: https://doi.org/10.1145/2736277.2741627

[25] S. Mahdavi, S. Khoshraftar, and A. An, "dynnode2vec: Scalable dynamic network embedding," in *2018 IEEE International Conference on Big Data (Big Data)*. Los Alamitos, CA, USA: IEEE Computer Society, dec 2018, pp. 3762–3765. [Online].

Available: https://doi.ieeecomputersociety.org/10.1109/BigData.2018.8621910

[26] C. D. T. Barros, M. R. F. Mendonça, A. B. Vieira, and A. Ziviani, “A survey on embedding dynamic graphs,” *ACM Comput. Surv.*, vol. 55, no. 1, nov 2021. [Online]. Available: https://doi.org/10.1145/3483595

[27] M. Belkin and P. Niyogi, “Laplacian Eigenmaps for Dimensionality Reduction and Data Representation,” *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, 06 2003. [Online]. Available: https://doi.org/10.1162/089976603321780317

[28] L. Torres, K. S. Chan, and T. Eliassi-Rad, “GLEE: Geometric laplacian eigenmap embedding,” *Journal of Complex Networks*, vol. 8, no. 2, mar 2020. [Online]. Available: https://doi.org/10.1093%2Fcomnet%2Fcnaa007

[29] K. Devriendt and P. V. Mieghem, “The simplex geometry of graphs,” *CoRR*, vol. abs/1807.06475, 2018. [Online]. Available: http://arxiv.org/abs/1807.06475

[30] M. Fiedler, *Matrices and Graphs in Geometry*, ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2011.

[31] L. Yao, L. Wang, L. Pan, and K. Yao, “Link prediction based on common-neighbors for dynamic social network,” *Procedia Computer Science*, vol. 83, pp. 82–89, 2016, the 7th International Conference on Ambient Systems, Networks and Technologies (ANT 2016) / The 6th International Conference on Sustainable Energy Information Technology (SEIT-2016) / Affiliated Workshops. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050916301259

[32] S. M. Kazemi, R. Goel, K. Jain, I. Kobyzev, A. Sethi, P. Forsyth, and P. Poupart, “Representation learning for dynamic graphs: A survey,” *Journal of Machine Learning Research*, vol. 21, no. 70, pp. 1–73, 2020. [Online]. Available: http://jmlr.org/papers/v21/19-447.html

[33] M. Haddad, C. Bothorel, P. Lenca, and D. Bedart, “TemporalNode2vec: Temporal Node Embedding in Temporal Networks,” in *COMPLEX NETWORKS 2019 : 8th International Conference on Complex Networks and their Applications*, Lisbon, Portugal, Dec. 2019. [Online]. Available: https://hal.science/hal-02332080

[34] A. Grover and J. Leskovec, “Node2vec: Scalable feature learning for networks,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 855–864. [Online]. Available: https://doi.org/10.1145/2939672.2939754

[35] M. Yang, M. Zhou, H. Xiong, and I. King, “Hyperbolic temporal network embedding,” *IEEE Transactions on Knowledge & Data Engineering*, vol. 35, no. 11, pp. 11 489–11 502, nov 2023.

[36] M. Kitsak, I. Voitalov, and D. Krioukov, “Link prediction with hyperbolic geometry,” *Phys. Rev. Res.*, vol. 2, p. 043113, Oct 2020. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.2.043113

[37] R. Ghrist and H. Riess, “Cellular sheaves of lattices and the tarski laplacian,” *Homology, Homotopy and Applications*, vol. 24, no. 1, pp. 325–345, 2022. [Online]. Available: https://doi.org/10.4310%2Fhha.2022.v24.n1.a16

[38] Z. Cooperband and R. Ghrist, “Towards homological methods in graphic statics,” 2023.

[39] I. H. R. Yoon and R. Ghrist, “Persistence by parts: Multiscale feature detection via distributed persistent homology,” 2020.

[40] S. MacLane, *Categories for the Working Mathematician*. New York: Springer-Verlag, 1971, graduate Texts in Mathematics, Vol. 5.

[41] F. Loregian, *(Co)end Calculus*. Cambridge University Press, jun 2021. [Online]. Available: https://doi.org/10.1017%2F9781108778657

[42] J. Curry, “Dualities between cellular sheaves and cosheaves,” 2016.

[43] J. L. Martin, “Geometry of graph varieties,” *Transactions of the American Mathematical Society*, vol. 355, pp. 4151–4169, 2003. [Online]. Available: https://api.semanticscholar.org/CorpusID:13118456

[44] P. Bubenik, M. Hull, D. Patel, and B. Whittle, “Persistent homology detects curvature,” *Inverse Problems*, vol. 36, no. 2, p. 025008, jan 2020. [Online]. Available: https://doi.org/10.1088%2F1361-6420%2Fab4ac0

[45] A. Testa, F. Farina, and G. Notarstefano, “Distributed submodular minimization via block-wise updates and communications,” *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2678–2683, 2020, 21st IFAC World Congress. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896320306728

[46] R. Ahlswede, N. Cai, S.-Y. Li, and R. W. Yeung, “Network information flow,” *IEEE Transactions on information theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[47] S.-Y. R. Li, R. W. Yeung, and N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[48] J. Widmer and J.-Y. Le Boudec, “Network coding for efficient communication in extreme networks,” in *Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking*, ser. WDTN ’05. New York, NY, USA: ACM, 2005, p. 284–291. [Online]. Available: https://doi.org/10.1145/1080139.1080147

[49] X. Zhang, G. Neglia, J. Kurose, D. Towsley, and H. Wang, “Benefits of network coding for unicast application in disruption-tolerant networks,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1407–1420, 2012.

[50] Z. Li, D. Zeng, S. Guo, S. Lu, D. Chen, and W. Zhuang, “On the throughput of feedbackless segmented network coding in delay tolerant networks,” *IEEE Wireless Communications Letters*, vol. 1, no. 2, pp. 93–96, 2012.

[51] F. Vieira, S. Shintre, and J. Barros, “How feasible is network coding in current satellite systems?” in *2010 5th Advanced Satellite Multimedia Systems Conference and the 11th Signal Processing for Space Communications Workshop*, 2010, pp. 31–37.

[52] P. Chen, Z. Xie, Y. Fang, Z. Chen, S. Mumtaz, and J. J. P. C. Rodrigues, “Physical-layer network coding: An efficient technique for wireless communications,” *IEEE Network*, vol. 34, no. 2, pp. 270–276, 2020.

[53] R. Wang, R. Ma, G. Liu, W. Kang, W. Meng, and L. Chang, “Joint link adaption and resource alloca-

tion for satellite networks with network coding," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2023.

[54] Z. Li and B. Li, "Network coding: The case of multiple unicast sessions," in *Allerton Conference on Communications*, vol. 16, no. 8, 2004.

[55] ——, "Network coding in undirected networks," in *Proc. 38th Annu. Conf. Int. Sci. and Sys.(CISS 2004)*, 03 2004, pp. 257–262.

[56] N. J. Harvey, R. D. Kleinberg, and A. R. Lehman, "Comparing network coding with multicommodity flow for the k-pairs communication problem," MIT, Tech. Rep. MIT-LCS-TR-964, 2004.

[57] M. Adler, N. J. A. Harvey, K. Jain, R. Kleinberg, and A. R. Lehman, "On the capacity of information networks," in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, ser. SODA '06. Society for Industrial and Applied Mathematics, 2006, p. 241–250.

[58] N. Harvey, R. Kleinberg, and A. Lehman, "On the capacity of information networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2345–2364, 2006.

[59] M. Braverman, S. Garg, and A. Schvartzman, "Coding in undirected graphs is either very helpful or not helpful at all," in *8th Innovations in Theoretical Computer Science Conference*, ser. ITCS 2017, 2017.

[60] P. Afshani, C. B. Freksen, L. Kamma, and K. G. Larsen, "Lower Bounds for Multiplication via Network Coding," in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, vol. 132, Dagstuhl, Germany, 2019, pp. 10:1–10:12. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2019/10586

[61] R. Ghrist and Y. Hiraoka, "Network codings and sheaf cohomology," in *2011 International Symposium on Nonlinear Theory and its Applications*, ser. NOLTA 2011. Kobe, Japan: Citeseer, 09 2011.

[62] ——, "Applications of sheaf cohomology and exact sequences to network coding," *Frontiers in mathematical science through collaborations with other disciplines*, vol. 1752, pp. 31–40, 07 2011.

[63] J. Curry, "Sheaves, cosheaves and applications," Ph.D. dissertation, University of Pennsylvania, 2013.

[64] R. W. Ghrist, *Elementary applied topology*. Createspace Seattle, 2014, vol. 1.

[65] E. W. Chambers, J. Erickson, and A. Nayyeri, "Homology flows, cohomology cuts," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, ser. STOC '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 273–282. [Online]. Available: https://doi.org/10.1145/1536414.1536453

[66] R. Ghrist and S. Krishnan, "A topological max-flow-min-cut theorem," in *2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 815–818.

[67] S. Krishnan, "Flow-cut dualities for sheaves on graphs," *arXiv preprint arXiv:1409.6712*, 2014.

[68] W. Fang, "A study of the multiple-unicast network coding conjecture using riemannian manifolds," 2016.

APPENDICES

PROOF OF PROPOSITION 3.1

Recall that we wish to show that a sequence of maps $f^n : C^nG_{i+1}, \mathcal{P}^{\mathbb{F}}_{i+1} \to C^nG_i, \mathcal{P}^{\mathbb{F}}_i$ is a cochain map for $n = 0, 1$, meaning $df^0 = f^1 d$. It is sufficient to prove the proposition for an arbitrary vertex v and an arbitrary section $s \in \mathcal{P}^{\mathbb{F}}_{i+1}(v)$

First, suppose $s \notin \mathcal{P}^{\mathbb{F}}_i(f^{-1}(v))$. Then,

$$\begin{aligned} d_i f^0(s) &= d_i 0 = 0 \\ f^1 d_{i+1}(s) &= f^1 \left(\sum_{e \mid v \le e} [v:e] \mathcal{P}^{\mathbb{F}}_{i+1}(v \le e)(s) \right) \\ &= \sum_{e \in G_i \mid v \le e} [v:e] f^1 \left(\mathcal{P}^{\mathbb{F}}_{i+1}(v \le e)(s) \right) \\ &= 0. \end{aligned}$$

Indeed, if $v \notin G_i$, then the last equality is trivial: without v there can be no edges incident to it. If $v \in G_i$, then we still have $f^1 \left(\mathcal{P}^{\mathbb{F}}_{i+1}(v \le e)(s) \right) = 0$ for all e such that $v \le e$. After all, $s \notin \mathcal{P}^{\mathbb{F}}_i(f^{-1}(v))$ means that either an outgoing edge or ingoing edge of v is not in G_i. Thus, the only situation in which $s_e = \mathcal{P}^{\mathbb{F}}_{i+1}(v \le e)(s) \ne 0$ is when there is an outgoing edge e that exists in G_i but an ingoing edge that does not (or vice versa). However, in this case, we must have that $s_e = \perp = 0$.

Now, suppose $s \in \mathcal{P}^{\mathbb{F}}_i(v)$. Then,

$$\begin{aligned} d_i f^0(s) = d_i(s) &= \sum_{e \mid v \le e} [v:e] \mathcal{P}^{\mathbb{F}}_i(v \le e)(s) \\ f^1 d_{i+1}(s) &= f^1 \left(\sum_{e \mid v \le e} [v:e] \mathcal{P}^{\mathbb{F}}_{i+1}(v \le e)(s) \right) \\ &= \sum_{e \in G_i \mid v \le e} [v:e] f^1 \left(\mathcal{P}^{\mathbb{F}}_{i+1}(v \le e)(s) \right) \end{aligned}$$

Now, the only sections $\mathcal{P}^{\mathbb{F}}_{i+1}(v \le e)(s)$ that do not get sent to 0 are the $\mathcal{P}^{\mathbb{F}}_{i+1}(v \le e)(s)$ such that $\mathcal{P}^{\mathbb{F}}_{i+1}(v \le e)(s) \in \mathcal{P}^{\mathbb{F}}_i(f^{-1}(e)) \implies \mathcal{P}^{\mathbb{F}}_{i+1}(v \le e)(s) \in \mathcal{P}^{\mathbb{F}}_i(e)$. In this case, then, we must have that $\mathcal{P}^{\mathbb{F}}_{i+1}(v \le e)(s) = \mathcal{P}^{\mathbb{F}}_i(v \le e)(s)$. That is,

$$f^1 d_{i+1}(s) = \sum_{e \in G_i \mid v \le e} [v:e] \mathcal{P}^{\mathbb{F}}_i(v \le e)(s) = d_i f^0(s).$$

In conclusion, we have shown that $d_i f^0 = f^1 d_{i+1}$.

PROOF OF PROPOSITION 5.2

Proof. Let $p = (e_0, e_1, \ldots, e_n)$ be a path in G. We construct a cochain x^p whose stalk data at v is the singleton flow $\{(e_i, e_{i+1})\}$ when v is a transshipment node and whose stalk data is $\{e_0\}$ or $\{e_n\}$ when v is the source or target respectively. It is easy to check that this cochain is a global section and that the mapping $p \mapsto x^p$ is injective.

To see this map is surjective, we will first show that for every $v \in V$ and $a \in \mathrm{Atom}(\Gamma(G; \mathcal{P}))$, we have $|a_v| \leq 1$. Suppose for a contradiction that there is some v such that $|a_v| > 1$ and assume v is a transshipment node without loss of generality. Then there are distinct flows $f, f' \in a_v$. Since a is a global section, we can inductively define a path from using f as follows. Initialize a path segment

$$(\dots, f_s, v, f_t, \dots)$$

where $f_s = (v_{-1}, v)$ and $f_t = (v, v_1)$. Since a is a global section there is some edge or flow $\phi_{-1} \in x_{v_{-1}}$ such that $\mathcal{F}_{v_{-1} < f_s}(\phi_{-1}) = \mathcal{F}_{v < f_s}(f) = \{e\}$ and similarly a $\phi_1 \in x_{v_1}$. We may then append these vertices and edges to each side of the path segment to form

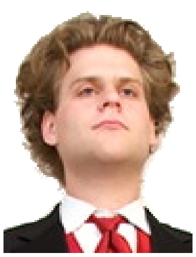
$$(\dots, (\phi_{-1})_s, v_{-1}, f_s, v, f_t, v_1, (\phi_1)_t, \dots).$$

Continue this process until the source and target are reached, which one can show will always happen even when G is not connected. This new path can viewed as a nonempty sub-global section of a, contradicting the fact a is an atom. Now given an atom a, the path mapping to a can be inductively constructed from source to target and the fact that $|a_v| \leq 1$ ensures this process is well defined.

Finally we show x^p is atomic. Suppose $y < x^p$ is a sub-global section so that there is some vertex u such that $y_u \subset x^p_u$. It follows that $y_u = \emptyset$ and hence $\mathcal{P}_{u \leq e}(y_u) = \emptyset$ for incident edges e of u. A simple induction argument shows we must have $y = \emptyset$ and hence x^p is atomic.

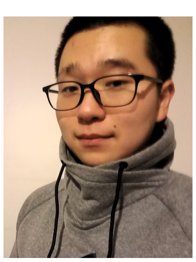
$\square$

Biography



Alan Hylton *would rather work in tube audio, but instead directs Delay Tolerant Networking (DTN) research at NASA GSFC. After studying mathematics at Cleveland State University and Lehigh University, he now advocates for his students and is humbled to work with his powerful and multidisciplinary team by creating venues for mathematicians to work on applied problems.*

Brendan Mallery *Brendan is a third year Ph.D. student studying mathematics at Tufts University. His research interests include optimal transportation and the study of Markov chains. Prior to Tufts he obtained a Masters in mathematics at the University at Albany, SUNY and a bachelor's in chemistry and mathematics at Bowdoin College.*



Jihun Hwang *(Jimmy) is a third-year Ph.D. student in computer science at Purdue University. He is primarily interested in information-theoretic cryptography and secure (multi-party) computations, but he ultimately likes to talk about any topics in or related to theoretical computer science and computer security. Before Purdue, he studied mathematics and computer science at the University of Massachusetts Amherst.*

Mark Ronnenberg *earned a Ph.D. in mathematics from Indiana University in 2023, where he was trained in gauge theory and low dimensional topology. He is now an assistant professor of math at Anne Arundel Community College. Outside of math, Mark loves music, books, and video games.*

Miguel Lopez *received his bachelor's degree in mathematics at Boston University and is currently a fourth-year Ph.D. student in applied math at the University of Pennsylvania. Under the supervision of Robert Ghrist, he is studying how algebraic topology can inform network science and machine learning algorithms. Outside of research Miguel is an avid board gamer and rock climber.*

Oliver Chiriac *received his B.Sc. in mathematics from the University of Toronto and is currently a M.Sc. student studying mathematics at the University of Oxford. Prior to this, he has done research in symplectic geometry and quantum field theory. He is interested in applying differential geometry and topology to the realms of physics and artificial intelligence. Oliver devotes his free time to soccer, music, and travel.*

Sriram Gopalakrishnan *is currently a second-year Ph.D. student at Sorbonne Université, LIP6, PolSys and University of Waterloo, SCG. His current research centers on the design and analysis of Gröbner basis algorithms to solve determinantal polynomial systems. He obtained his undergraduate degree in mathematics from the University of Utah and his MS in mathematics from Leiden University and Université de Bordeaux through the ALGANT program.*

Tatum Rask *is a fourth-year mathematics Ph.D. student at Colorado State University. Her research interests are in applied topology: specifically, she is interested in using tools from category theory and combinatorics to study persistent homology. Her main objective is to use pure math concepts to study applied problems. Outside of mathematics, Tatum enjoys trail running and reading.*